LNCS 2040

Weidong Kou
Yelena Yesha
Chung Jen Tan (Eds.)

Electronic Commerce Technologies

Second International Symposium, ISEC 2001
Hong Kong, China, April 2001
Proceedings

Springer

Lecture Notes in Computer Science 2040
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Weidong Kou Yelena Yesha
Chung Jen Tan (Eds.)

Electronic Commerce Technologies

Second International Symposium, ISEC 2001
Hong Kong, China, April 26-28, 2001
Proceedings

Springer

Volume Editors

Weidong Kou
The University of Hong Kong
E-Business Technology Institute
and
IBM China/Hong Kong
E-mail: wdkou@eti.hku.hk

Yelena Yesha
The University of Maryland Baltimore County
Department of Computer Science and Electrical Engineering
E-mail: yeyesha@cs.umbc.edu

Chung Jen Tan
The University of Hong Kong
E-Business Technology Institute
and
IBM T.J. Watson Research Centre
E-mail: ctan@eti.hku.hk

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Topics in electronic commerce : second international symposium ; proceedings / ISEC 2001, Hong Kong, China, April 26 - 28. 2001. Weidong Kou ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2040)
ISBN 3-540-41963-2

CR Subject Classification (1998): K.4.4, K.6.5, E.3, C.2, D.4.6, H.2.7

ISSN 0302-9743
ISBN 3-540-41963-2 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10782515 06/3142 5 4 3 2 1 0

Preface

The second International Symposium on Electronic Commerce was held in Hong Kong, April 2001, in conjunction with the fourth International Workshop on the Technological Challenges of Electronic Commerce. This symposium belongs to the e-commerce conference series started in 1998 in Toronto, Canada. Since then, every year, there has been an international workshop on the technological challenges of electronic commerce, and every other year, in conjunction with the workshop, an international symposium on electronic commerce is held. The following workshops have been held so far.

- The first International Workshop on the Technological Challenges of Electronic Commerce was held in September 1998, in Toronto, Canada.
- The second International Workshop on the Technological Challenges of Electronic Commerce was held in May 1999, in Beijing, China.
- The third International Workshop on the Technological Challenges of Electronic Commerce was held in June 2000, in Waterloo, Canada
- The fourth International Workshop on the Technological Challenges of Electronic Commerce was held in April 2001, in Hong Kong.

The first International Symposium on Electronic Commerce was held in Beijing, China, May 1999, in conjunction with the second International Workshop on the Technological Challenges of Electronic Commerce.

The second International Symposium on Electronic Commerce was sponsored by IBM, the E-Business Technology Institute and the Department of Computer Science and Information Systems at the University of Hong Kong, the Institute for Global Electronic Commerce at the University of Maryland Baltimore County, and Hong Kong Productivity Council. The goal of the symposium was to provide a forum for researchers, software vendors, and application developers and business users to share and disseminate information about current important research and application issues concerning electronic commerce. The symposium featured invited talks given by leading experts, presentations of refereed papers, and workshops and tutorials covering the major areas of electronic commerce. The additional goal of the symposium was to indicate the importance of electronic commerce technologies in the global marketplace including the fast growing Asia-Pacific market.

The papers accepted by the symposium program committee were gathered for preparing the proceedings. Among these papers, there are 8 regular papers, 4 short papers, and 2 survey papers. The authors were from Canada, the United States, Germany, Australia, Singapore, Hong Kong, and China, respectively. These papers covered various aspects of electronic commerce, including secure payment, trust and security, tools for e-commerce applications, and e-commerce frameworks and systems. There were also two e-commerce survey papers, one on B2B frameworks and standards, and the other on online auctions.

The staff at the E-Business Technology Institute at the University of Hong Kong were mainly responsible for organizing the symposium and preparing the proceedings. We are grateful to them, especially to Drs. William Song and Joshua Huang.

We would like to thank the members of the program committee for their efforts in organizing the review processes. Our thanks go to the reviewers who gave generously of their time to read and evaluate the papers. We would also like to thank the members of the Steering Committee for their guidance. We especially thank the conference organizers for the work they put into making this conference a successful one. We would like to thank the conference sponsors, particularly IBM China Limited/Hong Kong and IBM Centre for Advanced Study, for their support. Finally, we thank the authors of all submitted papers, in particular the accepted ones, the invited speakers, tutorial instructors, workshop session chairs and speakers, and all the participants who contributed to the success of the symposium.

April 2001

Weidong Kou
Yelena Yesha
Chung Jen Tan

Second International Symposium on Electronic Commerce, ISEC 2001

Sponsored by
E-Business Technology Institute
and
Department of Computer Science and Information Systems
University of Hong Kong

Institute for Global Electronic Commerce
University of Maryland Baltimore County

Hong Kong Productivity Council

International Business Machines Corporation

General and Program Co-chairs:

Weidong Kou	*University of Hong Kong*
Yelena Yesha	*University of Maryland Baltimore County*

Steering Committee:

Chung Jen Tan	*Director, ETI, The University of Hong Kong*
Gabby Silberman	*Director, IBM CAS, USA*
Francis Lau	*Head, CSIS, The University of Hong Kong*
George Wang	*Director, IBM China Research and Development Labs*
K.T. Yong,	*General Manager of IT, HKPC, Hong Kong*
Johnny Wong,	*Director, ICR, University of Waterloo, Canada*
Paul Timmer	*Director, E-Commerce, European Commission*

Program Committee:

Nabil Adam	*Rugters University, USA*
Neil Anderson	*Copenhagen Business School, Denmark*
Leo Liu	*IBM, USA*
David Cheung	*University of Hong Kong*
Lucas Hiu	*University of Hong Kong*
Dawn Jutla	*Saint Mary's University, Canada*
Jiandong Li	*Xidian University, China*
Monty Newborn	*McGill University, Canada*
Tamer Ozsu	*University of Waterloo, Canada*
T. Radhakrishnan	*Concordia University, Canada*
William Song	*University of Hong Kong*
B. Svedheim	*Framcom , Sweden*
Daniel Tan	*Nanyang Polytechic, Singapore*
Graham Williams	*CSIRO, Australia*
Yixian Yang	*Beijing Univ. of Posts and Telecom., China*
Jih-Shyr Yih	*IBM Research, USA*

Publication:

Joshua Huang — *University of Hong Kong*

Treasure:

William K.P. Chan — *ICO Limited, Hong Kong*

Workshops and Tutorials:

Jiming Liu — *Baptist University, Hong Kong*
William Song — *University of Hong Kong*

Local Arrangement and Sponsorship:

William Song — *University of Hong Kong*
Shirley Chow — *University of Hong Kong*
Mary Law — *IBM China/Hong Kong*

Referees:

Agnew, Gordon
Bo, Meng
Chang, Henry
Chen, Shyh-Kwei
Cheung, David
Chiasson, Theodore
Chung, Jen-Yao
Cooper, James W.
Edwards, Keith H.
Fader, Chris
Fu, Ada
Gate, Carrie
Hawkey, Kirstie
Ho, Wai Shing
Huang, Joshua
Hui, Lucas
Hui, Sui
Kontogiannis, Kostas
Kou,Weidong
Kwok, Michael
Lau, Terry
Lee, Juhnyoung
Lee, Sau Dan
Li, Jing
Li, Zichen
Litoiu, Marin
Liu, Y.C.
Lutfiyya, Hanan
Mamas, Evan
Molenkamp, Gary
Moser, Simon
Ng, Michael K.
Nguyen, Khanh
Radhakrishnan, T.
Rouse, Jason
Sans, Oda
See, Teo Loo
Shepherd, Michael
Solonim, Jacob
Song, Ronggong
Song, William
Tan, Daniel
Tian, Zhong
Tong, C.H. Frank
Wang, Huaxiong
Wang, Lian
Wang, Xiaoyun
Watters, Carolyn
Williams, Graham
Wong, David
Wong, Johnny W.
Wu, QiuXin
Yiu, S.M.
Yiu, Siu Wai
Yu, X. Jeffrey
Zhong, Ming

Table of Contents

Performance and QoS

E-Commerce Surveys

An Efficient Multiple Merchants Payment Protocol for Secure Electronic Transactions Based on Purchase Consolidation

Oda Sans[1] and Gordon B. Agnew[2]

[1] Lehrgebiet Nachrichtentechnik, FernUniversität Hagen, Feithstr. 142 / TGZ, 58084 Hagen, Germany
oda.sans@fernuni-hagen.de,
[2] Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1,
gbagnew@engmail.uwaterloo.ca

Abstract. In this paper we propose two payment protocols between one customer and m out of n merchants in an e-mall environment based on the SET protocol. Instead of interacting with each merchant separately (traditional SET scenario) the customer performs only one transaction paying all merchants at once.
The first protocol is based on a trusted consolidator who acts on behalf of the merchants performing the cryptographic actions and distributing the information to the merchants. The computational costs of this protocol are $4m + 17$ 1024−bit exponentiations compared to $19n$ 1024−bit exponentiations in the traditional SET scenario. The communication overhead is reduced from $12m$ kbit to $5m + 8$ kbit.
The second protocol is based on a (n, k)−threshold scheme set up among the merchants which allows all sets of k merchants to perform a cryptographic action together. It does not require a trusted party. The computational costs of this protocol are $6m+2k+12$ 1024−bit exponentiations. The communication overhead is to $6m + 5k + 7$ kbit.

1 Introduction

In today's electronic malls or e-malls, a variety of shops is presented in one virtual location on the Web to provide a convenient shopping experience for the customer. In systems currently in use, the financial transactions are still performed separately for each shop. Both the customer and the payment gateway have to repeat the payment process several times performing similar steps and processing similar data. The shopping experience ceases to be user-friendly and the computational load and communication overhead is unnecessarily high.

Considering the unequal computational power of the participants in the transaction, this set-up is certainly not ideal. The consumer machine is the least powerful and the computational load should therefore be as low as possible. This is especially true, if the customer is using a smart card providing security for the transaction. The payment gateway is a powerful single machine or cluster

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 1–19, 2001.

of machines and processes several thousand transactions in a minute, however it has limitations and it is important to sustain a fast response time even in the face of the expected increase in e-commerce and during peak hours.

We propose, therefore, a multiple merchants payment protocol based on the SET payment protocol that allows the customer to pay for all items in the e-mall within one transaction even if they are from different merchants. This scheme has several advantages: it is more user-friendly and more efficient since the customer has to pay only once upon leaving the e-mall. The consumer's communication and computation is reduced to one payment process comparable with one traditional payment process in the single merchant situation. The communication between the merchants and the payment gateway is also reduced to one interaction except of the generation of the capture tokens which have to be merchant specific (see also 4.1 and 4.2). The computation costs of the m merchants involved in the purchase decrease from $7m$ 1024bit exponentiations to $2m + 7$.

We start our paper with a review of the SET Payment Protocol and its computation and communication costs in Sect. 2 and we continue with related work on more efficient SET implementations using signcryption schemes by Zheng et al. and Seo and Kim in Sect. 3.

In Sect. 4 two multiple merchants payment protocols are proposed based on different trust models: In the first scenario, a fully trusted e-mall consolidator interacts with the consumer on behalf of the merchants. In the second scenario, a $(k, n)-$threshold scheme is set up among the n merchants. It is based on a trust model in which all merchants trust each other in so far as that they believe that there are never more than $k - 1$ malicious merchants.

The computation and communication costs of both protocols are compared to the original SET Payment Protocol and the signcryption protocols where the customer makes single billing transactions with each merchant. In addition the security properties are evaluated.

2 The SET Protocol

The SET Standard was developed by MasterCard and Visa to secure credit card based transactions over open networks like the Internet [7]. Before the introduction of SET, credit card payments over the Internet were handled in a manner similar to traditional Mail Order / Telephone Order purchases (MOTO) which are called *card-not-present*-transactions. This kind of transactions provides no authentication between merchants and customer if it is not provided elsewhere and led to major security concerns on merchants' and customers' side. This was perceived as a threat to the development of electronic commerce and especially to the use of credit cards in it. Therefore, the objectives of the SET protocol as described in the business plan [7] are:

– confidentiality of the payment instructions
– authentication of the parties involved

– integrity of order information and payment instructions
– interoperability

The SET protocol takes place between a customer (C), a merchant (M), and a payment gateway (PG) which is linked to a private financial network using proprietary protocols as shown in Fig. 1.

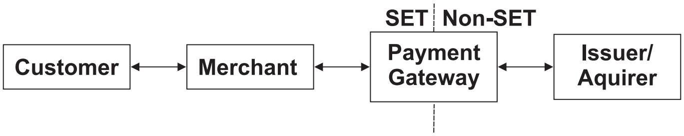

Fig. 1. Scope of the SET Protocol

2.1 Payment Transaction in SET

The SET protocol guarantees authenticated communications between the players and therefore customer, merchant, and payment gateway must have valid and certified public key pairs for digital signatures, $(pk_C^{(s)}, pv_C^{(s)})$, $(pk_M^{(s)}, pv_M^{(s)})$, and $(pk_{PG}^{(s)}, pv_{PG}^{(s)})$, and for encryption, $(pk_M^{(e)}, pv_M^{(e)})$ and $(pk_{PG}^{(e)}, pv_{PG}^{(e)})$, The generation and certification of a public key pair is also part of the SET protocol and takes place before any other transactions can be performed. In this paper, we assume that all parties completed this initial step successfully and concentrate on the actual payment protocol.

In the purchase process, SET does not come into play until the customer has decided on what she wants to buy and initiates the payment transaction. The detail of the order, e.g. product descriptions and amounts, have already been exchanged between merchant and customer during the online shopping process.

In the description of the protocols, we restrict our attention to the cryptographically relevant actions and the related data items. For a detailed description we refer to the SET specifications [8]. The data related to the customer is the order information (OI), and the payment instructions (PI); the data related to the merchant is financial information (FI); the data related to the payment gateway is the capture information ($CapToken$). All data related to a transaction ($TransID$) or a request/response pair ($RRPID$) is omitted.

Figure 2 shows an overview of the central purchase transaction between customer (C), merchant (M), and payment gateway (PG), as described in [5] and [8].

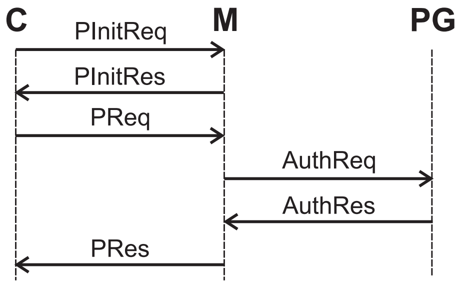

Fig. 2. Message flow in SET

1: C $\rightarrow$ M $PInitReq$: C initializes the purchase protocol.
2: C $\leftarrow$ M $PInitRes = \{S(PInitReq)\}$: M answers with the signed PInitReq
C checks the signature.
3: C $\rightarrow$ M $PReq = \{OI, H(OI), E_{pk^{(e)}_{PG}}(PI), H(PI), SO(H(H(OI), H(PI)))\}$: C sends the order information (OI), its hash, the payment instructions (PI) encrypted with the payment gateway's public encryption key, its hash, and the dual signature.
M checks the integrity of OI and the dual signature.
4: M $\rightarrow$ PG $AuthReq = \{H(OI), E_{pk^{(e)}_{PG}}(PI), SO(H(H(OI), H(PI))), E_{pk^{(e)}_{PG}}(S_{pv^{(s)}_{M}}(FI))\}$: M passes the encrypted payment instructions, the hash of the order information, and the dual signature to the payment gateway and sends the signed and encrypted financial information.
PG decrypts the payment instructions, checks the dual signature, verifies the authorization of the customer, decrypts the financial information and checks the signature.
5: M $\leftarrow$ PG $AuthRes = \{E_{pk^{(e)}_{M}}(S(AuthReq)), E_{pk^{(e)}_{PG}}(S(CapToken))\}$: PG sends the signed and encrypted AuthRes and the signed and with his own public key encrypted and signed CapToken for later use by M.
M decrypts the AuthRes, checks the signature, and stores the encrypted and signed CapToken.
6: C $\leftarrow$ M $PRes = \{S(PRes)\}$: M sends the signed PRes

2.2 Security Analysis

Our security analysis is based on the objectives of the SET protocol given in Sect. 1. First the communication is authenticated among all three parties by means of signature and encryption certificates. Second, data integrity and linkage is guaranteed by signed hashes or a dual signature in the case of the PReq step.

The privacy of the consumer's payment instructions is guaranteed by asymmetric encryption with the public encryption key of the payment gateway. Only the order information between customer and merchant travels in plain text. The privacy of the authentication messages, AuthReq and AuthRes, are both encrypted asymmetrically with the public encryption key of the receiving party.

The dual signature used in the PReq message allows both the merchant and the payment gateway to verify the linkage between order information and payment instructions without learning the content of the message received by the other party, and through the linkages in the AuthRes and PRes message, the merchant and the customer can verify the linkage of these messages to the whole transaction and the previously exchanged information. The verification can be performed by anybody knowing the public key of the sending party, and thus cheating of one of the parties can therefore be proved to a third party.

2.3 Efficiency Analysis

The SET protocol is based on 1024bit RSA, SHA-1 hash, and DES encryption combined with a RSA based digital envelope. Since the RSA operation requires much more computations than hashing or symmetric encryption[1] and since hashing and encryption is the same for all protocols, we calculate the computational costs based on the costs for RSA operations occurring in the form of digital signatures and digital envelopes for symmetric encryption.

The computational costs of a RSA operation can be calculated with regard to the expected number of multiplications it requires. This measurement of computational cost was also used in the papers on signcryption schemes by Zheng et al. [11] and Seo et al. [9] presented in Sect. 3 and therefore allows us to use their results immediately.

In the SET protocol, the RSA scheme is used with a 1024-modulus. Since there are no further restrictions to the selection of the private and public key, for the analysis we assume that both are 1024-bit numbers. Therefore, the generation of a signature (DSG) and its verification (DSV) require a 1024-bit exponentiation $m^x \bmod n$ with $m, x, n \in \mathbb{Z}_n$ which costs $\frac{1}{4}\frac{3}{2}1024 = 384$ 1024bit-multiplications using the square-and-multiply algorithm and Garners speed-up (see note 14.75 in [6]). The communication overhead of a digital signature is 1024bit. The public key encryption (E) and decryption (D) also require one exponentiation costing 384 1024-bit multiplications; the communication overhead consists of the 1024bit for the digital envelope of the symmetric encryption.

The overall number of multiplications and the communication overhead of the six steps of the SET payment protocol for the three parties are given in Table 1.

[1] In [3] it is estimated that DES is 100 times faster than RSA if implemented in software and up to 10,000 times faster if implemented in hardware.

Table 1. Computation and Communication cost of the SET Payment Protocol (number of 1024-multiplications / bit) (E=Encryption, D=Decryption, DSG=Digital Signature Generation, DSV=Digital Signature Verification)

	Comp. Cost			Comm. overhead
	C	M	PG	
1 PInitReq	–	–	–	–
2 PInitRes	– 1DSV=384	1DSG=384 –	– –	1024bits –
3 PReq	1DSG+1E=768 –	– 1DSV=384	– –	2048bits –
4 AuthReq	– –	1DSG+1E=768 –	– 2DSV+2D =1536	4096bits –
5 AuthRes	– –	– 1DSV+1D=768	2DSG+2E =1536 –	4096bits –
6 PRes	– 1DSV=384	1DSG=384 –	– –	1024bits –
$\sum = 7296$	1536	2688	3072	12288bits

3 Related Work

There have been two approaches to make SET more efficient with respect to the computation and communication costs based on new ways to link data and perform a dual signature: The LITESET scheme was proposed by Hanaoka, Zheng and Imai in 1998 [2] and is based on an earlier work of Zheng [11] where he introduced a signcryption scheme that combines encryption and digital signature of a message into one operation. The SlimSET scheme was developed by Seo and Kim in [9] and [10] and is based on a domain-verifiable signcryption scheme which extends the dual signature to a multiple signcryption scheme with a group of recipients. Both ideas reduce computation costs and communication overhead of the SET protocol.

3.1 The Signcryption Scheme by Zheng (1997)

In [11] a signcryption scheme is defined as a pair of (polynomial time) algorithms (S, U), where S is called the signcrypt algorithm and U the unsigncrypt algorithm. (S, U) fulfils the following three properties:

1. Unique Unsigncryptability: If a text m is signcrypted to $c = S(m)$ then $U(c)$ recovers m unambiguously.
2. Security: The pair (S, U) fulfils both the requirements of a signature scheme and an encryption scheme, i.e. confidentiality of the message, verifiable integrity of the message, and nonrepudiation.

3. Efficiency: The computational costs and the communication overhead is smaller than in a scheme concatenating signature and encryption (so called *signature-then-encryption scheme*).

Zheng proposed the following scheme based on a shortened DSS-signature scheme (SDSS1[2]). Alice has the public key pair $(x_a, y_a = g^{x_a})$ and Bob $(x_b, y_b = g^{x_b})$, the following parameters are publicly available: p a large prime, q a large divisor of $p-1$, g an element of $\{1, ...p-1\}$ of order q, KH a keyed hash function, (E, D) a pair of encryption and decryption algorithm.

– Signcryption by the sender Alice (A):
  1. A picks a random $x \in \{1, ...q\}$ and calculates $k = y_b^x \bmod p$ and splits k into an encryption key part k_1 and a signature key part k_2.
  2. A calculates the signature $r = KH_{k_2}(m)$.
  3. A generates $s = \frac{x}{r+x_a}\ q$.
  4. A encrypts the text $c = E_{k_1}(m)$.
  5. A sends Bob the tuple (c, r, s).
– Unsigncryption by the receiver Bob (B):
  1. B recovers $k = (y_a \cdot g^r)^{sx_b} \bmod p$ and splits it into k_1 and k_2.
  2. B decrypts $m = D_{k_1}(c)$.
  3. B accepts m if $KH_{k_2}(m) = r$

In [11] this scheme was compared to a signature-then-encryption scheme using Schnorr's signature scheme and ElGamal's encryption. The savings in computation costs were found to be 50% and over 75% in communication overhead, with the savings in the communication overhead going up to 96% with increasing size of the parameters.

In [2] the signcryption scheme was applied to the SET payment protocol with the security parameters p a 1024-bit number and q a 160-bit number. The following two message types were developed in order to be deployed in the SET messages.

Linked Data: $LinkedData_{A,y_b}(m_1, m_2) = \{r, s, c\}$ shows that a message m_1 is linked to message m_2 for a message going from sender A to receiver B with public key y_b. The message parts r, s, c are computed as follows:
  1. A picks a random $x \in \{1, \ldots, q-1\}$ and computes a pair of keys $(k_1, k_2) = H(y_b^x \bmod p)$.
  2. A calculates the joined hash $r = KH_{k_1}(H(m_1), H(m_2))$ and $s = \frac{x}{r+x_a} \bmod q$.
  3. A encrypts m_1 to $c = E_{k_2}(m_1)$.
  4. A sends (c, m_2, r, s)

  Bob's verification consists of the following three steps:
  1. B recovers the key pair $(k_1, k_2) = H((y_a \cdot g^r)^{sx_b} \bmod p)$.
  2. B decrypts $m_1 = D_{k_2}(c)$.
  3. B accepts the message if $r = KH_{k_1}(H(m_1), H(m_2))$.

[2] He proposed two similar schemes from which we use the first one in order to keep the following simple.

LinkedData can be applied to the *AuthReq* and the *AuthRes* messages in the SET protocol. The generation of *LinkedDate* requires 240 1024-bit multiplications, the verification 280, and it causes 320bit communication overhead with the given parameters.

Coupled Data: Coupled Data is used in the *PReq* message in place of the Dual Signature of the original SET scheme. In the paper two implementations are given: In the first one, both messages are encrypted and the message basically consists of two parts of LinkedData going to receiver B_1 and B_2, therefore the generation costs are 480 1024-bit multiplications, the verification for each recipient 280 multiplications and the message overhead is 640.
In the second implementation, only the message for receiver B_2 is encrypted and the message for the first receiver B_1 is only concatenated with a hash of both messages.

For the remaining messages in the SET protocol, *PInitRes* and *PRes*, the signcryption scheme SDSS1 is directly applied. The computation costs are 240 1024-bit multiplication for message generation and 280 for verification, the communication overhead is 320bit. In table 2 the savings for the three parties are given.

Table 2. Comparison of SET and LITESET (number of 1024-bit multiplications / bits)

	Comp. Cost				Comm. overhead
	$\sum$	C	M	PG	
SET	7296	1536	2688	3072	12288bits
LITESET	3360	1040	1280	1040	2560bits

The security properties of the LITESET scheme are similar to the SET scheme except of the identification of dishonest parties which requires the private key of the receiver and needs more computations than in the SET protocol. Besides, the signature of the coupled data is different for both receivers, so unlike the SET scheme the two receivers can not be confident about the correctness of the signature received by the other party even if their own signature proved to be valid.

3.2 The Domain-Verifiable Signcryption Scheme by Seo and Kim (1999)

In [9] Seo and Kim present the domain-verifiable signcryption scheme which allows to send a message m consisting of n parts $m_1, \ldots m_n$ to n recipients $B_1, \ldots B_n$ where each recipient can only decrypt his part of the message but can verify the signature of the whole transaction.

The public parameters of the scheme are: p a large prime, q a large prime factor of $p-1$, g an element of order q, H a hash function, HK a keyed hash function, (E, D) a pair of encryption and decryption function. The sender Alice has the public key pair $(x_a, y_a = g^{x_a} \bmod p)$, all recipients B_i have an own public key pair $(x_{b_i}, y_{b_i} = g^{x_{b_i}} \bmod p)$. The scheme follows the following steps:

- Signcryption by the sender Alice (A):
  1. A chooses a random $x \in Z_q^*$ and calculates the partial keys $k_i = H(y_{b_i}^x \bmod p)$ and the overall key $k = H(g^x \bmod p)$.
  2. A encrypts the parts of the message with the corresponding partial key $c_i = E_{k_i}(m_i)$.
  3. A calculates the partial signatures $r_i = KH_k(c_1 \parallel \ldots \parallel c_{i-1} \parallel m_i \parallel c_{i+1} \parallel \ldots \parallel c_n)$.
  4. A computes the digital envelope $s = \frac{x}{r_1 \cdots r_n + x_a} \bmod q$.
  5. A sends $(c_1, c_2, \ldots, c_n, r_1, r_2, \ldots, r_n, s)$ to each of the B_i's.
- Unsigncryption by the recipient B_i
  1. B_i computes $t = (y_a g^{r_1 \cdots r_n})^s \bmod p$ and his partial key $k_i = H(t^{x_{b_i}} \bmod p)$ and the overall key $k = H(t)$.
  2. B_i decrypts his part of the message $m_i = D_{k_i}(c_i)$.
  3. B_i accepts if $KH_k(c_1 \parallel \ldots \parallel c_{i-1} \parallel m_i \parallel \ldots \parallel c_{i+1} \parallel \ldots \parallel c_n) = r_i$.

This scheme is now adapted to show linkage between two messages in the following three ways:

$Link - I_{(A,B)}(m_1, m_2)$**:** Link-I provides a scheme that links m_1 to m_2 and sends it encrypted to the recipient B. Link-I corresponds to the LinkedData in Sect. 3.1. The generation of a Link-I message costs 480 1024-bit multiplications, its verification costs 1040 and it causes 480bit communication overhead.

$Link - II_{(A,B_1)}(m_1, m_2)$**:** Link-II provides a scheme where a message is send first to B_1 and from there to B_2 and B_1 should not be able to see the content of message m_2 but should be able to verify the linkage. Link-II corresponds to the CoupledData of Zheng and the DualSignature in the original SET protocol. The generation takes 480 1024-bit multiplications, the verification 560; the communication overhead is 400bit.

$Link - III_{(A,B_1,B_2)}(m_1, m_2)$**:** Link-III is the same as Link-II but encrypts the message m_1 also.

In the SlimSET protocol Link-I is used in the *AuthReq* and the *AuthRes* messages and Link-II is used in the *PReq* message. As in LITESET, SDSS1 is used for *PInitRes* and *PRes*. The computational costs of this scheme are somewhat higher for the merchant and the payment gateway than in the scheme proposed by Zheng et al., but it has even lower communication costs as is shown in Table 3.

The advantage of this scheme is, that the discovery of dishonest parties does not require the private keys of another party. In addition, the recipients share one signature and can therefore be confident about the correctness of the signature of the other parties.

Table 3. Comparison of SET and SlimSET (number of 1024-bit multiplications / bits)

	Comp. Cost				Comm. overhead
	$\sum$	C	M	PG	
SET	7296	1536	2688	3072	12288bits
SlimSET	4320	1000	1800	1520	2160bits

4 Multiple Merchants Payment Protocol

The above schemes provide us with efficient solutions for single SET transactions. In a e-mall, however, the customer interacts not only with one merchant but with a group of n merchants as shown in Fig. 3.

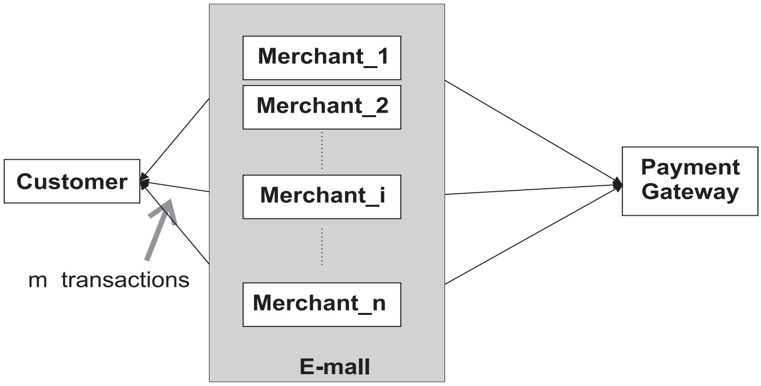

Fig. 3. Multiple Merchant Payment Scenario

After the shopping process, the customer might want to buy products from m different merchants and therefore has to go through the whole payment process for each of the m merchants separately. No matter how efficient the single transaction is made, this leads to a slow and complex situation and to an unsatisfactory shopping experience as shown in Table 4. This also imposes unnecessary workload on the payment gateway whose fast response is a crucial element of the payment transaction.

The objectives of a multiple merchants payment protocol are:

– The client has only to interact with one merchant or trusted third party to order all his purchases.
– Computation costs and communication overhead are reduced.
– The security level remains the same as in SET.

In addition, we want the customer side and the payment gateway to change as little as possible in order to allow interoperability with single transactions,

Table 4. Efficiency in the Multiple Merchant Scenario (in numbers of multiplications)

	# crypt. ops.	SET	LITESET	SlimSET
C	4	$m \cdot 1536$	$m \cdot 1040$	$m \cdot 1000$
M	7	$m \cdot 2688$	$m \cdot 1280$	$m \cdot 1800$
PG	8	$m \cdot 3072$	$m \cdot 1040$	$m \cdot 1520$
$\sum$	19	$m \cdot 7296$	$m \cdot 3306$	$m \cdot 4320$
overhead		$m \cdot 12288$bits	$m \cdot 2560$bits	$m \cdot 2160$bits

i.e., the software on the customer's side, the SET wallet, and on the payment gateway's side do not have to distinguish between a standard transaction and a joint transaction. However, order information obtained during the shopping process before the transaction may be adjusted, e.g., in order to identify the product with the merchant.

4.1 Scenarios 1: Consolidator as Trusted Third Party

In our first model an e-mall consolidator is introduced who acts on behalf of the merchants in the e-mall.

Trust Model. This model assumes that the merchants have a long-term relationship with the e-mall provider who is responsible for the e-mall consolidator. The merchants fully trust the consolidator who checks the certificates of the customer and the payment gateway, verifies the signature, and forwards the order information obtained from the customer. The consolidator passes the payment instructions to the payment gateway on behalf of the merchants and passes the CapTokens from the payment gateway back to the merchants.

The e-mall consolidator has his own public key pairs for signing $(pk^{(s)}_{MC}, pv^{(s)}_{MC})$ and encryption and $(pk^{(e)}_{MC}, pv^{(e)}_{MC})$.

The Payment Protocol. Before the actual purchase protocol is started the following changes have to be made within the shopping process: instead of the merchant's identity the identity of the consolidator is communicated to the customer. The order information is adapted in so far as all products belong to exactly one merchant, therefore the order information in the wallet has the format $OI = (OI_1, \ldots, OI_n)$ and the payment instructions $PI = \{PI_1, \ldots PI_n\}$ and two new steps are introduced to distribute the parts to the appropriate merchant (OISplit and CapSplit). The message flow is illustrated in Fig. 4.

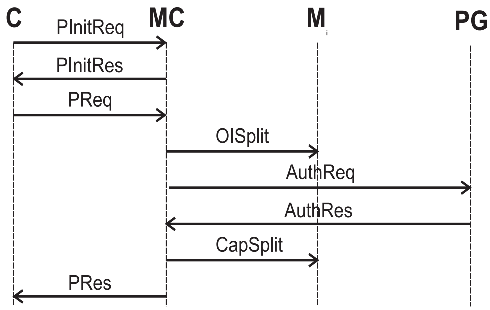

Fig. 4. Message Flow in the Consolidation Scenario

1: C $\rightarrow$ MC *PInitReq*: C initializes the purchase protocol with MC.
2: C $\leftarrow$ MC $PInitRes = \{S(PInitReq)\}$: MC signs the request and sends it back together with his certificate and the certificate with the encryption key of the payment gateway.
3: C $\rightarrow$ MC $PReq = \{OI, H(OI), E_{pk^{(e)}_{PG}}(PI), H(PI), SO(H(H(OI), H(PI)))\}$: C sends PReq to MC
4: MC $\rightarrow$ M_i $OISplit_i = \{S(OI_i)\}$: MC checks the dual signature, splits OI into its parts and sends the signed parts to the merchants M_i
5: MC $\rightarrow$ PG $AuthReq = \{H(OI), E_{pk^{(e)}_{PG}}(PI), SO(H(H(OI), H(PI))), E_{pk^{(e)}_{PG}}(FI_1, \ldots, FI_m)\}$: MC passes PI, the hash of OI, and the dual signature to PG and sends a signed and encrypted list of the financial information of the merchants.
6: MC $\leftarrow$ PG $AuthRes = \{E_{pb^{(e)}_{MC}}(AuthReq), E_{pk^{(e)}_{PG}}(S(CapToken_i))\}$: PG decrypts the payment instructions, checks the dual signature and checks the customer account. He sends the signed and encrypted *AuthReq* and m CapTokens encrypted with his own public key.
7: MC $\rightarrow$ M_i *CapSplit*: MC forwards the CapTokens included in the response of PG to the merchants
8: MC $\leftarrow$ C *PRes*: MC sends a signed return to the customer.

Security Properties. In this protocol the e-mall consolidator acts on behalf of the m merchants and the steps are similar to the original SET payment protocol. The security properties are the same as in the original SET protocol since the merchant is substituted by the e-mall consolidator: authentication, data integrity, and non-repudiation is maintained under the assumption that the consolidator is fully trusted. He identifies dishonest consumers or invalid capture tokens.

The merchants themselves are not able to check the signature of the customer because the single merchant does not know the whole order information and because there is no relationship between the single parts OI_i and PI_i received by merchant M_i. The order information is not encrypted between the consolidator and the merchants, but the parts are signed to authenticate their origin from the consolidator.

Efficiency. An overview of the computation costs and the communication overhead is given in Table 5. The consumer has the same amount of computation and communication costs as in a payment transaction with only one merchant. The savings in computation costs on the consumer's side are therefore $\frac{m-1}{m}$. The payment gateway has m additional digital signatures and encryptions to compute compared to the single merchant protocol. Compared to m single transactions with each of the m merchants separately the payment gateway saves $\frac{1}{2}$ of the digital signatures and encryptions and $\frac{m-1}{m}$ of decryptions and signature verifications.

We now compare the workload of the consolidator and the m merchants against the workload of all m merchants in the single transaction case. In the consolidator model we have $m+3$ instead of $3m$ digital signatures, $m+2$ instead of $2m$ signature verifications, and 1 instead of m encryptions and decryptions. A detailed overview is given in Table 5.

Table 5. Computation and Communication Cost of Scenario 1 (in number of operations and bits), (E=Encryption, D=Decryption, DSG=Digital Signature Generation, DSV=Digital Signature Verification)

	Computational Cost				Comm. overhead
	C	MC	M_i	PG	
1 PInitReq	–	–	–	–	–
2 PInitRes (gen.)	–	1DSG	–	–	1024bits
	1DSV	–	–	–	–
3 PReq	1DSG+1E	–	–	–	2048bits
	–	1DSV	–	–	–
4 OISplit	–	m·DSG	–	–	$m \cdot 1024$bits
	–	–	m·DSV	–	–
5 AuthReq	–	1DSG+1E	–	–	4096bits
	–	–	–	2DSV+2D	–
6 AuthRes	–	–	–	$(m+1)$DSG $+(m+1)$E	$m \cdot 2048$bits
	–	1DSV+1D	–	–	–
7 CapSplit	–	–	–	–	$m \cdot 2048$bits
	–	–	–	–	–
8 PRes	–	1DSG	–	–	1024bits
	1DSV	–	–	–	

Example 1. To illustrate our results, we calculate the reductions in communication cost and communication overhead for the case $m = 10$. The following table gives the number of cryptographic actions per participant and the percentage of the amount compared to the number needed in SET. The communication overhead is calculated over all messages.

Comp. Cost	#	Scenario 1	m·SET
C	4	4 (10%)	40(100%)
M	$2m+7$	27(39%)	70(100%)
PG	$2m+6$	26(33%)	80(100%)
$\sum$	$4m+17$	57(30%)	190(100%)
Comm.		59,392(49%)	122880(100%)

Consolidator with Merchant Privacy. This scenario can be extended to provide merchant privacy by encrypting the order information between customer and consolidator and between the consolidator and the merchants. Therefore the PReq transaction and the OISplit transaction have to be altered.

In the Preq transaction, C encrypts the parts of the message with the public key of the consolidator. A label is attached to the parts of the messages which indicates to which merchant they belong. After having verified the hash the consolidator decrypts the symmetric encryption key for each part and encrypts it with the public key of the respective merchant and send the newly encrypted message to the merchant. In this scenario the consolidator can read all order information parts, but outsiders and other merchants can not. However this alteration yields to no advantages over the above scheme as to the number of encryptions and decryptions needed.

4.2 Scenario 2: Shared Group Signature

In this scenario the consolidator is replaced by two $(n, k)-$threshold scheme set up amongst the n merchants of the e-mall sharing the private keys of the signature key pair and the encryption key pair. The shared generation and computation of RSA based was described by Boneh and Franklin in 1997 [1]. The shared implementation of LITESET and SlimSET is so far only possible for $(n, 2)-$schemes which were described by Langford in 1996 [4].

Trust Model. In the $(n, k)-$threshold scheme any set of k merchants can perform a signature or a decryption together. Therefore, each merchant has to trust k of his fellow merchants to act trustfully on his behalf. Only up to $k - 1$ dishonest merchants can be tolerated. The verification of digital signatures and encryption can be performed by any of the merchant because only public keys of the other parties are needed.

Like the consolidator in the above model, the group of merchants (GM) has two public key pairs $(pk_{GM^{(s)}}, pv_{GM^{(s)}})$ and $(pk_{GM^{(e)}}, pv_{GM^{(e)}})$. The private

keys are shared amongst the merchants in a (n, k)-threshold scheme, i.e., each merchant has a private signature key share $pv^{(s)}_{GM_i}$ and a private encryption key share $pv^{(e)}_{GM_i}$ $(i = 1, \ldots n)$. This can either be set up by a trusted party or in a distributed way by the merchants themselves if a RSA based scheme is used [1].

Payment Protocol. In the shopping process the customer gets the identity of one of the merchants who coordinates the payment process. Let's assume this is merchant M_1. For authentication, the customer gets the identity of the group of merchants. The flow of the protocol is described in Fig. 5.

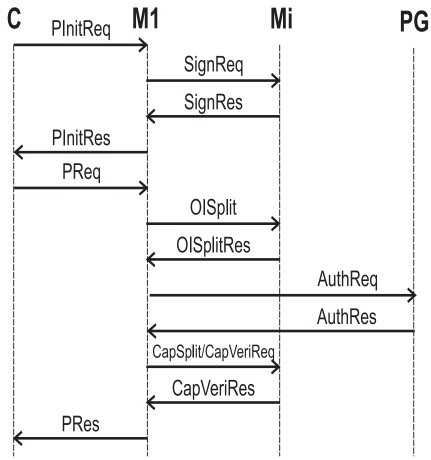

Fig. 5. Message Flow in the Threshold Scenario

The protocol steps are composed of the following elements:

1: C $\rightarrow M_1$ $PInitReq$: C initializes the purchase protocol with M_1.

2: $M_1 \rightarrow M_i$ $SignReq(H(PInitReq))$: M_1 computes the hash of the $PInitReq$ message, calculates his own partial signature $S_{pv^{(s)}_{GM_1}}(H(PInitReq))$, and sends $k-1$ $SignReq$ messages to other random selected merchants.

3: $M_1 \leftarrow M_i$ $SignRes = \{S_{pv^{(s)}_{GM_i}}(H(PInitReq))\}$: M calculates the group signature from the received partial signatures.

4: C $\leftarrow M_1$ $PInitRes = \{S(PInitReq)\}$: M_1 sends the signed $PInitReq$ message back to C.

5: C $\rightarrow M_1$ $PReq = \{OI, H(OI), E_{pk^{(e)}_{PG}}(PI), H(PI), SO(H(H(OI), H(PI)))\}$: C sends PReq to M_1.

6: $M_1 \rightarrow$ M$_i$ $OISplit_i = \{S(OI_i)\}$: M_1 checks the dual signature and splits OI into its parts and sends the signed parts to the m involved merchants.

7: $M_1 \leftarrow M_i$ $OISplitRes_i = \{E_{pk^{(e)}_{PG}}(S(FI_i))\}$: The m merchants answer with their financial information signed with their own private signature key and encrypted with the public encryption key of the payment gateway.

8: $M_1 \rightarrow$ PG $AuthReq = \{H(OI), E_{pk^{(e)}_{PG}}(PI), SO(H(H(OI), H(PI))), E_{pk^{(e)}_{PG}}(S(FI_1)), \ldots, E_{pb^{(e)}_{PG}}(S(FI_m))\}$: MC passes the PI and the dual signature and the list of signed and encrypted financial information of the merchants.

9: $M_1 \leftarrow$ PG $AuthRes = \{E_{pk^{(e)}_{GM}}(S(AuthReq)), E_{pk^{(e)}_{PG}}(CapToken_1), \ldots\}$: PG decrypts the payment instructions, checks the dual signature and checks the customer account. He sends the signed and encrypted $AuthReq$ and m CapTokens encrypted with his own public key.

10a: $M_1 \rightarrow$ M$_i$ $CapSplit_i = \{CapToken_i\}$: MC forwards the CapTokens included in the response of PG to $m - k$ of the merchants.

10b: $M_1 \rightarrow$ M$_i$ $CapVeriReq_i = \{CapToken_i, AuthRes, H(PRes)\}$: M_1 forwards the CapToken to the remaining k merchants and requests a partial decryption of the $AuthRes$ message and a partial signature of the $PRes$ message.

11: $M_1 \leftarrow M_i$ $CapVeriRes_i = \{D_{pv^{(e)}_{GM_i}}(AuthReq), S_{pv^{(s)}_{GM_i}}(H(PReq))\}$: the k merchants return the partially decrypted AuthRes and the partial signature of $H(PRes)$.

12: C $\leftarrow M_1$ $PRes = \{S_{pv^{(s)}_{MG}}(H(PReq))\}$: MC sends a signed return to the customer.

Security Properties. In the threshold scenario, privacy of the consumer data is guaranteed since the payment instructions are just passed by merchant M_1 processing the request. Even a collaboration of all merchants can not recover the information if we assume the underlying encryption algorithm to be secure.

The order information parts are not encrypted as in the original SET protocol. In addition, merchant M_1 explicitly receives all order information parts before passing them on. In order to prevent leakage of vital business information, the order information should be reduced to a minimum with the important information already exchanged during the shopping process. The financial information of the merchant in the AuthRes message are encrypted separately from each merchant involved in the transaction (step 7: OISplitRes) and is therefore secured in the same way as in the original SET transaction.

The most significant differences concerns the verification of digital signatures since this cryptographic action can be performed by the merchant M_1 alone. So we have to check if a collision of the consumer and the merchant M_1 can succeed in betraying the other merchants. First, they have to provide a valid dual signature since the payment gateway checks the dual signature. Second, they can not change the order information later-on since the payment gateway can always provide the original dual signature. So the betrayal is only possible with

the payment gateway's collaboration or with the merchant M_1 impersonating the payment gateway. In the first case, the other merchants can indeed be betrayed, but this is also true for the single merchant protocol. In the second case, the merchant M_1 has to fake the digital signature of the payment gateway as well as the encrypted CapTokens which is not possible under the assumption that RSA is secure.

Efficiency. The computation cost of the consumer is constant and equals the costs necessary for a single merchant transactions independent of the number of merchants involved. The computation costs of the payment gateway reduce from $2m$ to $m+1$ for each of the four cryptographic actions. The dependency from m is caused by the fact that the CapTokens have to be computed for each merchant individually.

The computation cost of the merchant for the 3 generations of digital signatures and the one decryption is reduced from $4m$ to $4k$, the cost for the one digital signature verification is reduced from m to 1. The number of encryptions remains n since each merchant encrypts his own financial data for privacy reasons (see section 4.2). For a detailed list of the computational costs see Table 6.

The communication overhead reduces from $12m \cdot 1024$bits to $(6m + 5k + 5) \cdot 1024$bits.

If only $1 < m < k$ merchants participate our scheme is less efficient for the merchants and the payment gateway than the m-fold invocation of the traditional scheme. It is still more efficient and more convenient for the customer.

Example 2. To illustrate our results, we give the savings of our protocol for a purchase involving $m = 10$ merchants and different thresholds $k = 2, \ldots, n$. We add all computation costs of all different cryptographic actions and all participants. The following table compares the computation costs with regard to the number of cryptographic actions and gives their percentage of the number of operations needed if the SET protocol is executed m times.

	#	$k=2$	$k=3$	$k=5$	$k=7$	$k=10$	n·SET
C	4	4 (10%)	4(10%)	4(10%)	4(10%)	4(10%)	40(100%)
M_I	$2m+3k$ $+3$	29(41%)	32(45%)	38(54%)	44(62%)	53(75%)	70(100%)
PG	$4m+4$	44(55%)	44(55%)	44(55%)	44(55%)	44(55%)	80(100%)
$\sum$	$6m+3k$ $+11$	77(40%)	80(42%)	86(45%)	92(48%)	101(53%)	190(100%)

To compare the communication overhead, we simply add the overhead of all messages. The following table gives the results in bits and the percentage:

$m=10$	$k=2$	$k=3$	$k=5$	$k=7$	$k=10$	n·SET
$6m+5k+7$	78,848 (64%)	83,968 (68%)	94,208 (76%)	104,448 (85%)	119,808 (97%)	122880 (100%)

Table 6. Computation and Communication Cost of Scenario 2 (in number of operations / bits) (E=Encryption, D=Decryption, DSG=Digital Signature Generation, DSV=Digital Signature Verification)

	Computational Cost				Comm. overhead
	C	M_1	M_i	PG	
1 PInitReq	–	–	–	–	–
2 SignReq	–	–	–	–	–
3 SignRes	–	1DSG	$(k-1)$DSG	–	$(k-1)$1024bits
4 PInitRes	– 1DSV	– –	– –	– –	1024bits –
5 PReq	1DSG +1E –	– 1DSV	– –	– –	2048bits –
6 OISplit	–	–	–	–	–
7 OISplitRes	– –	– –	mDSG + mE –	– –	$m \cdot 2048$bits –
8 AuthReq	– –	- –	– –	– $(m+1)$DSV +$(m+1)$D	$(m+1)$2048bits –
9 AuthRes	– –	– –	– –	$(m+1)$DSG +$(m+1)E$ –	$(m+1)$2048bits –
10 CapSplit	– –	– –	– –	– –	k·2048bits –
11 VeriRes	– –	– 1DSV	kD+kDSG –	– –	k·2048bits –
12 PRes	– 1DSV	1DSG –	– –	– –	1024bits

5 Conclusion

In the paper, we analyzed the situation of e-mall shopping where, after shopping with different merchants, a customer wants to pay for all items in one transaction. We proposed two scenarios based on two trust models with a trusted consolidator acting on behalf of the merchants and with a threshold scheme set up among the merchants.

Both protocols fulfill the goals of a multiple merchants protocol given in Sect. 4: they offer the same security as the original SET scheme and reduce computation cost and communication overhead. The reductions are most significant on the consumer's side whose computation costs are constant and independent of the number of merchants involved in the transaction. The savings for the merchants and the payment gateway are still dependent of the number of merchants involved (based on the fact that the CapTokens are computed individually) but they are reduced to a fraction of the computation costs needed in executing the SET protocol for each merchant separately.

Acknowledgements. This paper was supported by the Canadian Institute for Telecommunications Research and the German Academic Exchange Service.

We thank Peter Cresswell and Arthur Lam for their support, the discussions, and the enjoyable working environment.

Our special thanks receives Professor Kittel of the FernUniversität Hagen who encouraged this work and provided us with valuable support.

References

1. Boneh, D. and Franklin, M.: Efficient Generation of Shared RSA Keys. In: Burton, S. and Kalinski, J. (eds.): Advances in Cryptology - Proceedings of Crypto '97. Lecture Notes in Computer Science, Vol. 1294. Springer-Verlag, Berlin Heidelberg New York (1997) 425-439
2. Hanaoka, G., Zheng, Y. and Imai, H.: LiteSET: a Light-Weight Secure Electronic Transaction Protocol. In: Boyd, C. and Dawson, E. (eds.): Proceedings of the Conference on Information Security and Privacy. Lecture Notes in Computer Science, Vol. 1438. Springer-Verlag, Berlin Heidelberg New York (1998) 215-226
3. RSA Security Inc.: FAQ - Frequently Asked Questions about Today's Cryptography (Version 4.1), http://www.rsasecurity.com/rsalabs/faq, May (2000)
4. Langford, S. K.: Threshold DSS Signatures without a Trusted Party. In: Coppersmith, D.(eds.): Advances of Cryptology - Proceedings of Crypto '96. Lecture Notes in Computer Science, Vol. 963 . Springer-Verlag, Berlin Heidelberg New York (1996) 397-409
5. Loeb, L.: SET - Secure Electronic Transaction. Artech House Publisher, Bosten (1998)
6. Menezes, A. J., Oorschot, P. C., and Vanstone, S. A.: Handbook of Applied Cryptography. CRC Press LLC (1997)
7. MasterCard and Visa: SET Secure Electronic Transaction Specification - Book 1: Business Description, Version 1.0 (1997)
8. MasterCard and Visa: SET Secure Electronic Transaction Specification - Book 3: Formal Protocol Definition, Version 1.0 (1997)
9. Seo, M. and Kim, K.: Another Improved SET: SlimSET. In: Proceedings of the 2000 Symposium on Cryptography and Information Security (2000)
10. Seo, M. and Kim, K.: Electronic Funds Transfer Protocol Using Domain-Verifiable SignCryption Scheme. In: Song, J.-S. (ed.): Proceedings of the International Conference on Security and Cryptography (ICISC '99). Lecture Notes in Computer Science, Vol. 1787. Springer-Verlag, Berlin Heidelberg New York (2000) 269-277
11. Zheng, Y.: Digital Signcryption or How to Achieve Cost(Signature & Encryption) $<<$ Cost(Signature) + Cost(Encryption). In: Burton, S. and Kalinski, J. (eds.): Advances in Cryptology - Proceedings of Crypto '97. Lecture Notes in Computer Science, Vol. 1294. Springer Verlag, Berlin Heidelberg New York (1997) 165-179.

A Fair Electronic Cash Scheme

Yi Mu[1], Khanh Quoc Nguyen[2], and Vijay Varadharajan[1]

[1] Department of Computing, Macquarie University,
Sydney, NSW 2109, Australia
[2] Motorola Lab, Adelaide, Australia

Abstract. This paper proposes a fair anonymous electronic cash scheme that ensures fairness for both clients and vendors. The proposed scheme is based on the Nyberg-Rueppel digital signature scheme, therefore it provides an alternative for the construction of fair electronic cash. The proposed scheme meets all basic security requirements for fair electronic cash including fairness, anonymity, confidentiality, authenticity, efficiency and double-spending detection.

1 Introduction

In an e-cash scheme, users or clients enjoy unconditional privacy. There is no any mechanism for banks and vendors to identify a client in a transaction without breaking the underlying number theoretic assumptions. This protection, which is desirable from clients' viewpoints, is a major concern for law enforcement agencies. It was pointed out in [2,18] that anonymous e-cash can be a “safe haven” for criminal activities that include money laundering, illegal purchases, perfect blackmailing and other attacks. This prevents the deployment of anonymous e-cash systems in a large scale, where such attacks and many others are often expected.

Brickell, Gemmell, and Kravitz[2] suggested a solution to this problem using escrowed cash, often known as *fair electronic cash*. The main feature of fair e-cash is the existence of a trusted authority that can revoke the anonymity of any given coin. The trusted authority is referred as the revocation authority. A different and more efficient solution was later proposed by Camenisch, Piveteau, and Stadler [14]. Both solutions require the revocation authority to be actively involved in every withdrawal and thus are not desirable.

Frankel, Tsiounis, and Yung[12] and Camenisch, Maurer, and Stadler[6] proposed a fair e-cash scheme employing an off-line revocation authority. The advantage of this approach is that the revocation authority is not involved in any payment transaction. When needed, the revocation authority can be called upon to identify the owner of a coin or a transaction. The most efficient schemes to date are those by Davida, Frankel, Tsiounis, and Yung[9] and by Camenisch, Maurer, and Stadler[6]. Both of these two schemes are constructed from Brands' anonymous e-cash scheme.

In this paper, we propose a fair electronic cash scheme using an off-line revocation authority. The proposed fair electronic cash scheme is based on Nyberg-

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 20–32, 2001.

Rueppel digital signature scheme and thus poses as an alternative to Schnorr based fair electronic cash schemes.

The paper is organised as follows. Section 2 discusses the model of a fair e-cash system using an o -line revocation authority. Section 3 gives an overview of Nyberg-Rueppel digital signature scheme and its blind version. Section 4 gives a brief discussion of a previously proposed anonymous electronic cash scheme. Section 5 shows the construction of a fair electronic cash scheme from the e-cash scheme. Finally, Section 6 gives our concluding remarks.

2 Model of Fair Electronic Cash

A fair e-cash scheme consists of four main parties, a bank B, a trusted authority T, vendors, and users. Each of vendors and users has an account with the bank. For convenience, we denote by V a vendor and U a user.

A fair e-cash scheme consists of ve basic protocols, three of them are the same as in anonymous e-cash, namely a withdrawal protocol with which U withdraws electronic coins from B, a payment protocol with which U pays some coins to V, and a deposit protocol with which V deposits the coins to B. The two additional protocols are conducted between B and T, i.e., *owner tracing* and *coin tracing* protocols. They work as follows:

- In the owner tracing protocol, B gives to T the view of a deposit protocol and T returns a string that contains speci c information which allows B to identify the owner using the database of user accounts.
- In the coin tracing protocol, B gives to T the view of a withdrawal protocol and T returns some speci c information that allows B to identify the coin in the deposit phase.

These two additional protocols provide the revocation capacity and the protection against certain types of attacks. For instance, the owner tracing protocol allows the authorities to identify the origin of dubious coins and thus eliminates money laundering. The coin tracing protocol allows the authorities to nd the destination of dubious coins and thus eliminates blackmailing.

3 Preliminary

3.1 Nyberg-Rueppel Digital Signature Scheme

The key generation protocol works as follows. Let p be a large prime and q be equal to $p \otimes 1$ or a large integer factor of $p \otimes 1$. Also let g be a random generator of subgroup G_q over $\mathbb{Z}_p$. Each signer chooses $x \in_R \mathbb{Z}_q$ and computes $h = g^x$. The signer's secret and public keys are x and h respectively.

To sign a message $m \in \mathbb{Z}_p$, the signer selects a random number $w \in \mathbb{Z}_q$ and computes r and s as

$$r = mg^w \bmod p \text{ and } s = xr + w \bmod q.$$

The pair (r, s) is the signature of the message m. To verify the validity of a signature, one checks

$$m = g^{\otimes s} h^r r \bmod p.$$

This scheme provides message recovery. The signature needs not to be accompanied by the message m.

3.2 Blind Nyberg-Rueppel Digital Signature Scheme

To obtain a blind Nyberg-Rueppel digital signature on a message m from the signer, the veri er needs to get a pair (r', s') satisfying

$$m = g^{\otimes s'} h^{r'} r' \bmod p,$$

in such a way that the signer does not learn anything about either r' or s'. This is achieved using the procedure given in Figure 1.

Recipient		Signer
		$w \in_R \mathbb{Z}_q$ $a \leftarrow g^w$
	$\xleftarrow{\quad a \quad}$	
$\alpha, \beta \in \mathbb{Z}_q$ $r' \leftarrow mg^{\alpha} a^{\beta}$ $m' \leftarrow r'/\beta$		
	$\xrightarrow{\quad m' \quad}$	
		$s \leftarrow m'x + w \bmod q$
	$\xleftarrow{\quad s \quad}$	
$s' \leftarrow s\beta + \alpha \bmod q$		

Fig. 1. Blind Nyberg-Rueppel digital signature scheme

The pair (r', s') is then a blind signature on message m. The correctness of the signature is shown as follows:

$$\begin{aligned} g^{\otimes s'} h^{r'} r' &= mg^{\otimes s \quad \otimes \quad +xr'+w \quad +} \\ &= mg^{\otimes m'x \quad \otimes w \quad +r'x+w} \\ &= mg^{xr' \otimes xm'} \quad = m \pmod p. \end{aligned}$$

The blindness holds because if   and   are chosen at random, r' and m are uniformly distributed in their respective domains. As r' and m uniquely identify s', (r', s', m) is a random triple and independent of the signer's view.

So far no apparent security weakness of this protocol is known. Some security proofs of this protocol have been discussed in [5,17]. Particularly, [17] shows that the view of the signer in the protocol and the signature are statistically independent, i.e., generated signatures are witness-indistinguishable.

4 An Anonymous Electronic Cash Scheme

In this section, we review a previously proposed electronic cash scheme. The scheme is based on blind Nyberg-Rueppel digital signature scheme and was proposed in [15].

4.1 The Setup Protocol

On inputting a security parameter k, the bank B runs a key generation algorithm generating the following:

- a large prime p and a large number q such that $q|(p \otimes 1)$,
- three generators g, g_1 and g_2 of the unique subgroup G_q of the multiplicative group $\mathbb{Z}_p$,
- a randomly chosen collision-intractable hash function $\mathcal{H}()$ of polynomial size in k that maps its inputs to $\mathbb{Z}_q$,
- a random number $x \in \mathbb{Z}_q$ and
- there numbers h, h_1 and h_2 computed as $h = g^x, h_1 = g_1^x$ and $h_2 = g_2^x$ (all are computed under $\mathbb{Z}_p$).

B's secret key and public key are (x) and $(p, q, g, g_1, g_2, h,\ h_1, h_2, \mathcal{H}())$ respectively.

4.2 The Account Setup

To set up an account with B, a user U chooses a random $u \neq 0 \in \mathsf{G}_q$ and forms $I = g_1^u \bmod p$. B regards $I \neq 1$ as user U's account identification and sends $z = (Ig_2)^x \bmod p$ to U. Note that I is the unique link to the user's real name, while u is unknown to the bank. u can be computed by the bank only when the user double spends a coin.

4.3 The Withdrawal Protocol

The withdrawal protocol between U and B is run over an authenticated channel and given in Figure 2.

The blind Nyberg-Rueppel digital signature scheme is essential for anonymity. Note that the base used in the protocol is not a fixed base g of the signer public key, but is the base $(Ig_2)^t$ for a random number t chosen by the user. At the end of the withdrawal protocol, the user should receive the blind Nyberg-Rueppel signature $\mathrm{Sign}(A, Z) = (A, Z, z', a', r', s')$ which is verified using the following equation:

$$\mathcal{H}(A, Z, a', z') = (a'^{\otimes s'})(z'^{r'})(r').$$

It is important to verify that the secret key used in the signature generation is the secret key x of the bank. Otherwise, the user can create such a signature using any secret key. This verification is done at the payment protocol.

$$
\begin{array}{lcr}
\text{User}(\mathsf{U}) & & \text{Bank}(\mathsf{B}) \\
\hline
t, x_1, x_2 \in_R \mathbb{Z}_q^* & & w \in_R \mathbb{Z}_q \\
a' \leftarrow (Ig_2)^t & & a \leftarrow (Ig_2)^w \\
z' \leftarrow z^t & & \\
A \leftarrow g_1^{x_1} g_2^{x_2} & & \\
Z \leftarrow h_1^{x_1} h_2^{x_2} & & \\
 & \xleftarrow{\quad a \quad} & \\
m \leftarrow \mathcal{H}(A, Z, a', z') & & \\
\alpha, \beta \in_R \mathbb{Z}_q^* & & \\
r' \leftarrow m {a'}^{\alpha} a^{t\beta} & & \\
m' \leftarrow r'/\beta \bmod q & & \\
 & \xrightarrow{\quad m' \quad} & \\
 & & s \leftarrow m'x + w \bmod q \\
 & \xleftarrow{\quad s \quad} & \\
s' \leftarrow s\beta + \alpha \bmod q & &
\end{array}
$$

Fig. 2. The withdrawal protocol

4.4 The Payment Protocol

The payment protocol is run between U and a vendor V using an anonymous channel. The payment of a coin (A, Z, z', a', r', s') is described in Figure 3. As

$$
\begin{array}{lcr}
\text{User}(\mathsf{U}) & & \text{Vendor}(\mathsf{V}) \\
\hline
 & & c \leftarrow \mathcal{H}(\mathsf{V} \| Date \| Time \| \ldots) \\
r_1 \leftarrow c(ut) + x_1 \bmod q & \xleftarrow{\quad c \quad} & \\
r_2 \leftarrow ct + x_2 \bmod q & & \\
 & \xrightarrow{r_1, r_2,\ \mathrm{Sign}(A,Z)} & \mathcal{H}(A, Z, a', z') \stackrel{?}{=} {a'}^{-s'} {z'}^{r'} r' \\
 & & g_1^{r_1} g_2^{r_2} \stackrel{?}{=} {a'}^{c} A \\
 & & h_1^{r_1} h_2^{r_2} \stackrel{?}{=} {z'}^{c} Z
\end{array}
$$

Fig. 3. The payment protocol

for the proof of equality of discrete logarithms, for a random challenge c if

$$
\begin{aligned}
g_1^{r_1} g_2^{r_2} &\stackrel{?}{=} {a'}^{c} A \\
h_1^{r_1} h_2^{r_2} &\stackrel{?}{=} {z'}^{c} Z,
\end{aligned}
$$

we must have $\log_{a'} z' = \log_{g_1} h_1$. This shows that the bank's secret key $x = \log_{g_1} h_1$ was used in the generation of Sign(A, Z).

4.5 The Deposit Protocol

V can deposit the coin Sign(A, Z) at any suitable time. The deposit procedure is to send the transcript of the payment to B. B veri es the payment procedure and accepts the coin if V follows the procedure correctly and the coin satis es all veri cations as checked in the payment phase.

5 A Fair Electronic Cash Scheme

This section presents a fair e-cash scheme using the described anonymous e-cash scheme.

5.1 The setup

Let $x \in \mathbb{Z}_q$ be the secret key of B. The public key of B consists of tuple: $(p, q, g, g_1, g_2, g_3, h = g^x, h_1 = g_1^x, h_2 = g_2^x)$, where g, g_1, g_2, g_3 are generators selected from $\mathbb{Z}_p$, and h, h_1, h_2 are computed under modulo p.

The secret key and the account number of a user U are u and $I = g_1^u \bmod p$ respectively. The only addition is a trusted authority T whose secret and public keys are and $(g_3, h_{T1} = g_1 \bmod p, h_{T2} = g_2 \bmod p)$, respectively.

5.2 The withdrawal protocol

The withdrawal protocol is based on the withdrawal protocol of the anonymous e-cash scheme. It is run between U and B over an authenticate channel. Formally, the protocol is given in Figure 4.

The withdrawal protocol is essentially the blind Nyberg-Rueppel digital signature scheme. The base used is $d^t = (Ig_2)^t g_3$. At the end of the withdrawal protocol, the user should receive the blind Nyberg-Rueppel signature

$$\text{Sign}(A_1, A_2, Z) = (A_1, A_2, Z, z', a', r', s')$$

which is veri ed using the following equation:

$$\mathcal{H}(A_1, A_2, Z, a', z') = (a'^{\otimes s'})(z'^{r'})(r').$$

Also the bank stores the value e in the coin database for future references.

In this protocol, the value A is split into two values A_1 and A_2 . This is necessary to achieve owner tracing. The extra computation of (d, e) and the associate proof of knowledge

$$\mathsf{PK}\{(u, t) : d = g_1^u g_2 g_3^{1/t} \wedge e = h_{T1}^{ut} h_{T2}^t\},$$

is to achieve coin tracing, where we meant that the prover proves his knowledge on (u, t) from the given (d, e) without revealing the value of (u, t). This notation will also be used later on. The details of the proofs can be found in Appendix A.

User(U)		Bank(B)
$t, x_1, x_2 \in_R \mathbb{Z}_q^*$		$w \in_R \mathbb{Z}_q$
$d \leftarrow I g_2 g_3^{1/t}$		
$e \leftarrow h_{T1}^{ut} h_{T_2}^t$		
	$\xrightarrow{d,e}$	
$\mathsf{U} \Rightarrow \mathsf{B} : \mathsf{PK}\{(u,t) : d = g_1^u g_2 g_3^{1/t} \wedge e = h_{T1}^{ut} h_{T2}^t\}$		
$a' \leftarrow d^t$		$a \leftarrow d^w$
$z' \leftarrow z^t$		
$A_1 \leftarrow g_1^{x_1}$		
$A_2 \leftarrow g_2^{x_2}$		
$Z \leftarrow h_1^{x_1} h_2^{x_2}$		
	$\xleftarrow{a}$	
$m \leftarrow \mathcal{H}(A_1, A_2, Z, a', z')$		
$\alpha, \beta \in_R \mathbb{Z}_q^*$		
$r' \leftarrow m {a'}^{\alpha} a^{t\beta}$		
$m' \leftarrow r'/\beta \bmod q$		
	$\xrightarrow{m'}$	
		$s \leftarrow m'x + w \bmod q$
	$\xleftarrow{s}$	
$s' \leftarrow s\beta + \alpha \bmod q$		

Fig. 4. The withdrawal protocol

5.3 The payment protocol

The payment protocol is run between U and V using an anonymous channel. The payment of a coin $(A_1, A_2, Z, z', a', r', s')$ is described in Figure 5. The payment protocol is similar to the payment protocol of the anonymous e-cash scheme, whereas U now splits a' into $a_1 = I^t = g_1^{ut} \bmod p$ and $a_2 = g_2^t \bmod p$ and proves to V that

$$\mathsf{PK}\{(u, t, \ \ ) : a_1 = g_1^{ut} \wedge a_2 = g_2^t \wedge D_1 = g^u h_{T1} \wedge D_2 = g_1\}.$$

This proof is used to achieve user tracing.

5.4 The deposit protocol

V can deposit the coin $\mathrm{Sign}(A_1, A_2, Z)$ at any suitable time. The deposit procedure is to send the transcript of the payment to B. B veri es the payment procedure and accepts the coin if V follows the procedure correctly and the coin satis es all veri cations as checked in the payment phase.

User(U)		Vendor(V)
$\rho \in_R \mathbb{Z}_q$		$c \leftarrow \mathcal{H}(\mathsf{V} \| Date \| Time \| \ldots)$
$r_1 \leftarrow c(ut) + x_1$	$\xleftarrow{\quad c \quad}$	
$r_2 \leftarrow ct + x_2$		
$D_1 \leftarrow g^u h_{T_1}^{\rho}$		
$D_2 \leftarrow g_3^{\rho}$		
	$\xrightarrow{r_1, r_2, a_1, a_2, D_1, D_2, \mathrm{Sign}(A_1, A_2, Z)}$	
		Verify: $\mathrm{Sign}(A_1, A_2, Z)$
		$g_1^{r_1} \stackrel{?}{=} {a_1}^c A_1$
		$g_2^{r_2} \stackrel{?}{=} {a_2}^c A_2$
		$h_1^{r_1} h_2^{r_2} \stackrel{?}{=} {z'}^c Z$
		$a' \stackrel{?}{=} a_1 a_2 g_3$

$$\mathsf{U} \Rightarrow \mathsf{V} : \mathsf{PK}\{(u, t, \rho) : a_1 = g_1^{ut} \wedge a_2 = g_2^t \wedge D_1 = g^u h_{T1}^{\rho} \wedge D_2 = g_1^{\rho}\}$$

Fig. 5. The payment protocol

The completeness of the scheme is straightforward. As the scheme is developed using our proposed anonymous e-cash scheme, it is easy to show that this fair e-cash scheme satis es all security requirements of the anonymous e-cash scheme, e.g., unforgeability. It remains to show that user tracing and coin tracing can be satis ed.

5.5 Anonymity Revocation

There are two possible anonymity controls in this scheme. One is to identify the user in a payment transaction and the other is to identify the history, i.e., the life cycle of a coin. The former is referred as user tracing and the latter is referred as coin tracing. In practice, T should only run these protocols under a court order. Formally, the user tracing and coin tracing protocols work as follows:

5.6 User tracing

To identify the user in a payment transaction, B brings (D_1, D_2) to T. In turn, T computes

$$D_1/D_2 = g^u h_{T1}/g_1 \;\; = g^u \bmod p,$$

which identi es the user.

The soundness of this protocol is due to the proof of knowledge

$$\mathsf{PK}\{(u, t, \;\;) : a_1 = g_1^{ut} \wedge a_2 = g_2^t \wedge D_1 = g^u h_{T1} \wedge D_2 = g_1\} \bmod p,$$

that shows g^u is the plaintext corresponding to the ElGamal ciphertext (D_1, D_2) encrypted using T's public key for the user secret information u. In the user

tracing protocol, T simply decrypts the ciphertext and returns g^u which identi es the user.

Note that this procedure is not possible for other parties as only T can decrypt a ciphertext encrypted using T's public key.

5.7 Coin tracing

Identifying a coin history can be done in two di erent ways. One is to identify the coin payment for a given coin withdrawal and the other is to identify the coin withdrawal for a given coin payment.

In the latter case, B sends to T the payment transcript. Then T computes the value

$$a'/g_3 \;\; = ((Ig_2)^t) \;\; = g_1^{ut} \;\; g_2^t \;\; = h_{T1}^{ut} h_{T2}^t = e \bmod p,$$

and returns the value e back to the bank. The anonymity revocation is done by searching for the computed value e in the coin withdrawal reference database.

In the former case, B sends to T the withdrawal reference e. Then T computes and returns to B the value

$$e^{1/} \;\; g_3 = h_{T1}^{ut/} \;\; h_{T2}^{t/} g_3 = g_1^{ut} g_2^t g_3 = a' \bmod p.$$

Now, the anonymity revocation is done by matching this computed value a' with the value a' in every deposited coin.

6 Concluding Remarks

We have proposed a fair e-cash scheme based on an o -line revocation authority. It ensures that the payment processes run between clients and vendors are fair. The proposed fair e-cash scheme is based on the Nyberg-Rueppel signature scheme, therefore it provides a new method for the construction of fair e-cash. Our scheme and other computationally-e cient (fair) e-cash schemes use discrete logarithm problems as the underlying assumption. It will be interesting to design an e cient RSA-based fair e-cash scheme.

References

1. S. Brands. Rapid demonstration of linear relations connected by boolean operators. In *Advances in Cryptology -Eurocrypt'97*, pages 318–333. LNCS vol. 1223, Springer-Verlag, 1997.
2. E.F. Brickell, P. Gemmell, and D. Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *In symposium on Distributed Algorithms(SODA)*. Albuquerque, NM. Available from http://www.cs.sandia.gov/psgemme/., 1995.
3. J. Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problems*. PhD thesis, Swiss Federal Institute of Technology, Zurich, 1998.

4. J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes. Technical Report RS-98-29, BRICS, 1999. an abstract version appeared in Proceeding of Eurocrypt'99, LNCS vol. 1592, pages. 106–121.
5. J. Camenisch, J. M. Piveteau, and M. Stadler. Blind signatures based on the discrete logarithm problem. In *Advances in Cryptology – Eurocrypt'94*, pages 428–432. LNCS vol. 950, Springer-Verlag, 1995.
6. M. Camenisch, U. Maurer, and M. Stadler. Digital payment systems with passive anonymity-revoking trustees. In *ESORICS'96*, pages 33–43. LNCS vol. 1146, Springer-Verlag, 1996.
7. D. Chaum, J. Evertse, and J. van de Graaf. An improved protocol for demonstrating possession of discrete logarithms and some generalizations. In *Advances in Cryptology- EUROCRYPT'87*, pages 127–141. LNCS vol. 304, Springer-Verlag, 1988.
8. R. Cramer and T.P. Pedersen. Improved privacy in wallets with observers. In *Advances of Cryptology - Eurocrypt'93*, pages 329–343. LNCS vol. 765, Springer-Verlag, 1994.
9. G. Davida, Y. Frankel, Y. Tsiounis, and M. Yung. Anonymity control in e-cash. In *Proceedings of 1st Financial Cryptography conference*. LNCS vol. 1318, Springer-Verlag, 1997.
10. U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1:77–94, 1988.
11. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO'86*, pages 186–194. LNCS vol. 263, Springer-Verlag, 1987.
12. Y. Frankel, Y. Tsiounis, and M. Yung. Indirect discourse proofs: achieving fair off-line e-cash. In *Advances in Cryptology – ASIACRYPT'96*, pages 286–300. LNCS vol. 1163, Springer-Verlag, 1996.
13. E. Fujisaki and T. Okamoto. Statistical zero-knowledge protocols to prove modular polynomial relation. In *Advances in Cryptology – CRYPTO'97*, pages 16–30. LNCS vol. 1294, Springer-Verlag, 1997.
14. J. Piveteau J. Camenisch and M. Stadler. An efficient payment system protecting privacy. In *Computer Security - ESORICS'94*, pages 207–215. LNCS vol. 875, Springer-Verlag, 1994.
15. Khanh Quoc Nguyen, Yi Mu, and Vijay Varadharajan. A new digital cash scheme based on blind nyberg-rueppel digital signature. In *Information Security Workshop*, pages 312–320. LNCS vol. 1396, Springer-Verlag, 1997.
16. T. Okamoto. An efficient divisible electronic cash scheme. In *Advances in Cryptology – CRYPTO'95*, pages 439–451. LNCS vol. 963, Springer-Verlag, 1995.
17. M. Stadler. *Cryptographic Protocols for Revocable Privacy.* PhD thesis, Swiss Federal Institute of Technology, Zurich, 1996.
18. B. von Solms and D. Naccache. On blind signatures and perfect crimes. *Computer and Security*, 11(6):581–583, 1992.

A Proofs of Knowledge About Discrete Logarithms

This section provides protocols to prove the knowledge of secret keys for proving that they satisfy given predicates. All the protocols presented in this section are well-known and mainly borrowed from the excellent PhD thesis by Camenisch[3] which gives a rigorous treatment of proofs of knowledge about

discrete logarithms. The interactive versions of these protocols are known to be witness-indistinguishable and proofs of knowledge. The reader is also referred to [1,4,13,16] for detailed discussions of these protocols and other variations.

We rst present these protocols as interactive protocols between a prover and a veri er. Later we will discuss how to use the techniques introduced in [10,11] to turn these protocols into signature schemes.

In the following, we assume that $g, h, h_1, h_2, g_1, \ldots, g_m \in \mathsf{G}$ are the generators of $\mathsf{G}_q/\mathbb{Z}_p$ for some known prime numbers q and p such that computing a representation any generator with respect to other generators is infeasible.

A.1 Proving the Knowledge of Discrete Logarithms

A proof of knowledge of the discrete logarithm proves the knowledge of the discrete logarithm of a public key y to the base g. More formally, it is a protocol that proves the knowledge of a secret number x such that $y = g^x$. This protocol is in fact the Schnorr identi cation scheme. Following the the notations of [3,4], we denote this protocol as

$$\mathsf{PK}\{(\ \ ) : y = g\ \ \}.$$

A.2 Proving the Knowledge of a Representation

A proof of knowledge of a representation proves the knowledge of a representation of y to the bases $g_1, \ldots, g_k$ which is denoted as

$$\mathsf{PK}\{(\ _1, \ldots, \ _m) : y = \prod_{i=1}^{m} g_i^{\ \ i}\}.$$

This proof is rst introduced in [7] and is given in Figure 6.

Prover		Verifier
$(g_1, \ldots, g_m, q, y, x_1, \ldots, x_m)$		$(g_1, \ldots, g_m, q, y)$
$r_1, \ldots, r_m \in_R \mathbb{Z}_q$ $w \leftarrow \prod_{i=1}^{m} g_i^{r_i}$	$\xrightarrow{w}$	
		$c \in \{0,1\}^k$
	$\xleftarrow{c}$	
$s_i \leftarrow r_i - c x_i \bmod q \ (i = 1, \ldots, m)$	$\xrightarrow{s_1, \ldots, s_m}$	
		$w \stackrel{?}{=} y^c \prod_{i=1}^{m} g_i^{s_i}$

Fig. 6. A proof of representation of y to the bases $g_1, \ldots, g_m$

The correctness of this protocol is due to

$$y^c y^c \prod_{i=1}^{m} g_i^{s_i} = \prod_{i=1}^{m} g_i^{cx_i} \prod_{i=1}^{m} g_i^{s_i} = \prod_{i=1}^{m} g_i^{cx_i+s_i} = \prod_{i=1}^{m} g_i^{r_i} = w.$$

The soundness is due to the fact that given a same value w, if the prover can answer two di erent challenges c and c' correctly, the knowledge extractor obtains two sets of $(c, s_1, \ldots, s_m)$ and $(c', s'_1, \ldots, s'_m)$ and extract the secret x_i as:

$$x_i = \frac{s_i \otimes s'_i}{c' \otimes c} \bmod q.$$

The zero-knowledge holds because a honest veri er can construct a valid view by choosing $s_1, \ldots, s_m$ and c at random and computing $w = y^c \prod_{i=1}^{m} g_i^{s_i}$.

A.3 Proving the Equality of Discrete Logarithms

So far, we have only considered the knowledge of discrete logarithms and representations. An extension is to prove not only the knowledge of secret keys but also certain relations among them. The most basic relation is the equality relation.

In the most simplest form, it is a proof of knowledge and of equality of discrete logarithm of y_1 to the base g_1 and y_2 to the base g_2. This proof was rst introduced by Chaum and Pedersen in [8]. Let us denote this protocol as:

$$\mathsf{PK}\{(\ ) : y_1 = g_1\ \wedge y_2 = g_2\ \}.$$

The intuition is to run the proof of knowledge of discrete logarithm of y_1 to the base g_1 and of discrete logarithm of y_2 to the base g_2. Then $\log_{g_1}(y_1) = \log_{g_2}(y_2)$ only if the prover can return the same answer in both cases for a random challenge chosen by the veri er. Formally, this protocol is given in Figure 7.

Prover $(g_1, g_2, q, y_1, y_2, x)$		Verifier (g_1, g_2, q, y_1, y_2)
$r \in_R \mathbb{Z}_q$ $w_1 \leftarrow g_1^r$ $w_2 \leftarrow g_2^r$		
	$\xrightarrow{w_1, w_2}$	
		$c \in \{0,1\}^k$
	$\xleftarrow{c}$	
$s \leftarrow r - cx \bmod q$		
	$\xrightarrow{s}$	
		$w_i \stackrel{?}{=} y_i^c g_i^s \ (i = 1, 2)$

Fig. 7. A proof of $\log_{g_1}(y_1) \equiv \log_{g_2}(y_2)$

It is trivial to extend the proof system of equality of discrete logarithms to a proof system of equality of representations. One of such proof is the proof of knowledge of representation of y_1 and y_2 to the bases (g_1, h_1) and (g_2, h_2) respectively and that the representation of y_1 to g_1 and y_2 to g_2 are equal. This protocol which is denoted as

$$\mathsf{PK}\{(\ \ ,\ \ _1,\ \ _2) : y_1 = g_1\, h_1{}^{1} \wedge y_2 = g_2\, h_2{}^{2}\},$$

is described in Figure 8. This proof introduced in [8], is the basic building block for many blind digital signatures and anonymous electronic cash schemes.

Prover $(g_1, g_2, h_1, h_2, q, y_1, y_2, x, z_1, z_2)$		Verifier $(g_1, g_2, h_1, h_2, q, y_1, y_2)$
$r, \rho_1, \rho_2 \in_R \mathbb{Z}_q$ $w_1 \leftarrow g_1^r h_1^{\rho_1}$ $w_2 \leftarrow g_2^r h_2^{\rho_2}$		
	$\xrightarrow{w_1, w_2}$	
		$c \in \{0,1\}^k$
	$\xleftarrow{c}$	
$s \leftarrow r - cx \bmod q$ $\sigma_i \leftarrow \rho_i - cz_i\ (i = 1, 2)$		
	$\xrightarrow{s, \sigma_1, \sigma_2}$	
		$w_i \stackrel{?}{=} y_i^c g_i^s h_i^{\sigma_i}\ (i = 1, 2)$

Fig. 8. A proof of equality of representation of y_1 to g_1 and y_2 to g_2

A Secure Payment Protocol Using Mobile Agents in an Untrusted Host Environment

Amitabha Das and Yao Gongxuan

School of Computer Engineering, Nanyang Technological University,
Singapore 639798
asadas@ntu.edu.sg

Abstract. Mobile agents are believed to be playing an important role in future e-commerce systems, offering great flexibility and improved performance. Yet, their adoption is largely hampered by the new security issues they raise. Among them, the most difficult to solve is the issue of protecting mobile agents against malicious hosts. While no known general solution to this problem exists, solutions providing effective protection against specific threats from malicious hosts are possible. In this paper, we propose a secure payment protocol using mobile agents that protects the confidentiality of sensitive payment information from spying by malicious agents. The protocol makes use of Shamir's secret sharing scheme. The security properties of the protocol are proven, and an analysis of its message complexity is provided.

1 Introduction

Mobile agent architecture can bring many potential benefits such as great flexibility and improved performance into distributed systems. Therefore, they are believed to be playing an important role in future electronic commerce systems. Not only can they provide a very flexible approach for information gathering on prices and goods available from the several catalog servers they visit, but also they can effectively take over the different aspects of the electronic transaction, from price settlement to paying and delivery of the goods purchased.

There are mainly three stages of activities in any e-commerce system: Information gathering; Negotiation; Payment and Delivery. In the first stage, mobile agents can be used to gather information from several hosts representing stores. These are offers made in response to a query performed by the agent, which refers its owner's wishes("I want to buy a camera for less than $500"). In the negotiation stage, the final terms of the transaction are fixed. Mobile agents can be used to mediate in this part of the transaction. Finally, in the payment and delivery stage, mobile agents can be used to make payments for the products being bought and to collect a receipt as a proof.

However, mobile agents and agent based systems raise new security issues including host protection and agent protection[2,8]. Securing agents against attacks from malicious hosts is a new and difficult problem. This problem is especially serious in the payment scenario where the mobile agent may experience stealing

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 33–41, 2001.

resources attacks. In this attack the server visited by the mobile agent could try to steal the agent's con dential information such as electronic money or credit card number.

The problem of protecting a mobile agent against attacks by a malicious host is made particularly di cult by the fact that the host must of necessity have full access to the agent's code and state in order to execute it. There is no general solution to this problem at this moment. Several existing approaches so far[5,6] are far from mature enough to be used in real applications. This can be seen as a major limitation on the usefulness of these mobile agents for e-commerce.

In this paper we present a protocol for electronic payment using mobile agents that is secure against malicious hosts. This work is a part of the ongoing project on mobile agent based e-commerce system at the Centre for Advanced Information Systems at Nanyang Technological University, Singapore. The protocol makes use of the threshold scheme of Shamir[7,3] for secret sharing. In this protocol, more than one agent are used in order to perform a secure mobile agent based transaction. The valuable secret such as electronic money is distributed into several shares in a way that no single share can reveal any information about the secret. Each mobile agent carries a single share of the secret so that even if it is spied on by a malicious host the secret is not compromised. We also introduce a non-repudiation[10] mechanism into the protocol to prevent a dishonest merchant from denying that the electronic money has been received by him.

The paper is organized as follows. In the next section we discuss the model of e-commerce adopted for our system, and the assumptions about the security threats. In section 3, we provide an overview of the complete payment protocol, followed by a detailed discussion of the core secure payment protocol in section 4. A brief analysis of the protocol is presented in section 5, and concluding remarks are provided in section 6.

2 The Model of Mobile Agents Based E-Commerce System

The mobile agent based e-commerce system adopted by our research group consists of subsystems at the buyer sites, the seller (or merchant) sites, the directory sites and the broker sites, a bank and one or more trusted third parties(TTPs) that provide some trust service. The buyer, seller, directory and broker agents in each site interoperate in the system. A directory agent acts as a matchmaker that helps buyer agents locate seller agents as well as broker agents. A broker agent facilitates the transactions between buyer and seller agents. All the broker sites are capable of serving as mobile agent hosts (MA hosts), the others may or may not host mobile agents. Figure 1 shows the architecture of the system.

The Bank provides a set of basic banking services that includes checking accounts, lines of electronic cash and money transfer service. The banking system implemented in the system is similar to the electronic cash system developed by DigiCash, Inc. [9]. The TTP acts as a party trusted by both the buyer and the seller, and can be used to mediate for non-repudiation service.

In this paper we base our discussion on the e-cash based payment system [4, 9]. The mobile agents in the payment scenario carry digital cash with them, so

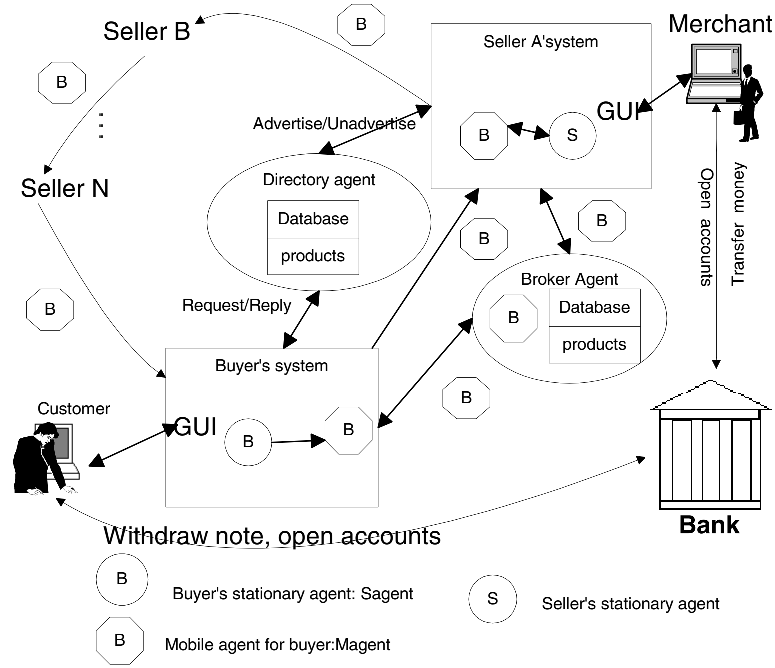

Fig. 1. The architecture of the mobile agent based e-commerce system

that e ective security measures are needed to protect mobile agents. However, the proposed protection scheme against malicious hosts is equally applicable to any other payment system.

2.1 The Threat Model

In this model, one or more agents representing a customer needs to transfer sensitive payment information to the merchant site. The customer agents are hosted by a network of MA hosts. Both the MA hosts and the merchant site are untrusted. However, the following assumptions apply to the nature of the potential attack on the customer agents:

- **A1.** At the most $m \otimes 1$ malicious hosts may collude to steal sensitive information from the customer agent.
- **A2.** The merchant may deny that he has received the electronic money, and spend the money as his own later.
- **A3.** The merchant site host works independently and does not collude with malicious hosts to cheat the customer.
- **A4.** There is a “trusted third party”(TTP) which is honest, and both the customer and the merchant trust that TTP will execute his role correctly.

The TTP is introduced to protect the customer against repudiation by the merchant and vice versa. The TTP has no role to play in the protection of the customer agents against malicious MA hosts.

3 The Secure Payment Protocol

3.1 Participants

There are mainly four roles in the payment system: a bank B, a customer C, a merchant M and a TTP. Both customer C and Merchant M have an account with the bank B. An electronic payment system consists of protocols that allow customer C to make a payment to the merchant M. TTP is used to provide non-repudiation service in case any party will deny sending or receiving information in the protocol.

3.2 Main Phases

There are mainly four phases in our payment system:
Withdrawal Phase. This phase involves the bank B and the customer C. We adopt this phase from the e-cash system here without modification.

The customer blinds the note number with a random value and sends it to the bank. The blind signature[1] used here is a very important tool that allows payments to be unconditionally untraceable by preventing the bank from recognizing the source of a deposited coin. The blinding can only be removed by the party who created it. The bank adds a digital signature to legitimize the "blinded" note, and the customer is able to unblind the note while maintaining the validity of the bank's digital signature. There are other components of the e-cash which are omitted here for brevity.
Distribution Phase. In this phase, the customer C encrypts the e-cash using a secret key and divides the secret key into several shares using a (m, n) threshold scheme. It creates 'n' mobile agents which are near replica of each other. Each of them carries the encrypted e-cash along with a distinct share of the secret key. After their creation, the customer dispatches them to different hosts.
Payment Phase. In this phase m mobile agents work together and each passes its share of the secret key to the merchant's site. The merchant then reconstructs the secret key after receiving m shares and deciphers the e-cash.
Verification and Transfer Phase. After the merchant reconstructs the *Money*, he signs the money with his private key and forwards it to the bank. The bank strips off the Merchant's signature, and verifies that the *money* has not already been spent by checking the serial number in its database. The bank adds funds corresponding to the note value to the merchant's account. It then prepares a signed receipt to the merchant who forwards it to the agent playing the leading role. This phase is very similar to that in the e-cash system.

4 The Core Secure Payment Protocol

The core protocol consists of the distribution phase, and the payment phase. The other two phases are very similar to those in e-cash systems.

4.1 Notations

- M: The Merchant(payee).
- s: Secret key for encrypting the e-cash.
- C: The electronic cash to be transferred as payment.
- $e_s(m)$: Message m symmetrically encrypted using secret key s.
- $d_s(g)$: Encrypted message g decrypted using secret key s.
- $(m)^k$: Message m asymmetrically encrypted with a public/private key k.
- $S_X(m)$: Message m digitally signed by X.
- Share$_i$: Share or shadow that the ith agent carries.
- $H(\ )$: One way hash function. It produces a xed-size output for any argument of any size. Given $H(x)$, it is computationally infeasible to determine x. It is also collision free, i.e., it is computationally infeasible to nd distinct x and y such that $H(x) = H(y)$.
- $H(i)$: Message digest for each share, $H(i) = H(share_i)$.
- *Lagent*: A leader agent used to organize the transfer of the shares, any of the mobile agents carrying a share of the secret can be a leader agent.
- P$_x$ and V$_x$: The public/private key pair of party x, where x= *Lagent*, M, or TTP.
- F_{REC}, F_{SUB}, F_{CON} : Flags used to identify the steps of transferring the shares in the protocol; they indicate the intended purpose of a (signed) message. F_{REC} indicates the objective of the step is transferring a receipt; F_{SUB} stands for submit whereas F_{CON} stands for con rm.
- TTP: On-line trusted third party providing security services accessible to the public.

4.2 Distribution Phase

The steps of this phase are as follows:

1. Distribute the Secret Key s Using Secret Sharing Scheme

The customer distributes s using the secret sharing scheme, a (m, n) -threshold scheme.

An arbitrary polynomial of degree $m \otimes 1$ is generated:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{m \otimes 1} x^{m \otimes 1}, \text{ where } a_0 = s$$

The coe cients $a_1, a_2, \ldots, a_{m \otimes 1}$ are chosen randomly from the set $Z_p = \{0, 1, \ldots, p \otimes 1\}$ and are kept secret and discarded after the shadows are handed out. p, is a prime larger than the number of possible shadows(shares), and the largest possible secret. All arithmetic will be done modulo p.

The n shadows are obtained by evaluating the polynomial at n random different points:

$$y_i = f(x_i)$$

The values of $x_i (i = 1, \ldots, n)$ are made public whereas $y_i (i = 1, \ldots, n)$ are kept secret and act as the n shares $Share_i (i = 1, \ldots, n)$. It is easy to see that when any m shadows come together, linear algebra can be used to solve for the coe cients of the polynomial including $a_0 = s$.

2. Compute n Message Digests for n Shares
A hash function $H(\ )$ is used to compute $H(i) = H(share_i), (i = 1, \ldots, n)$. Due to the special character of the hash function, if $H(x_1) = H(x_2)$, then $x_1 = x_2$, it can be used to judge if two inputs of the function are the same.
3. Dispatch n Mobile Agents to n Hosts
The customer creates n agents, each agent will carry the encrypted e-cash $e_s(C)$, a share of the secret and all the n message digests, $H(j)(j = 1, \ldots, n)$, $x_j(j = 1, \ldots, n)$ are added for reconstructing the secret later. The n agents are dispatched to n distinct hosts. So each mobile agent carries:

$$e_s(C), share_i, \{(x_j, H(j)), j = 1, \ldots, n\}$$

4.3 Payment Phase

The payment phase begins when one of the mobile agents of the customer initiates the payment process after receiving the payment order from the merchant. This agent will be designated as the leader agent or *Lagent*. The payment order, PO, consists of a number of components as given below:

$$\text{PO} = S_M(Tid, Lagent, M, Goods\_desc, Amount)$$

where, Tid = Unique identi er for the transaction being carried out,
$Goods\_desc$ = Description of the goods being purchased,
$Amount$ = The amount to be paid to the merchant,
$S_M(\ )$ = Indicates that the message is signed by M.

The *Lagent* will randomly select other $m \otimes 1$ mobile agents and send the payment order signed by the merchant. After other $m \otimes 1$ mobile agents have veri ed the payment order, they will send their shares to the merchant. The merchant sends a signed acknowledgement to *Lagent* after he has received $m \otimes 1$ shares. The *Lagent* then sends the last share to a TTP from which the merchant collects it.
1. Initialization
The *Lagent*, sends the following information to randomly selected $m \otimes 1$ other mobile agents: the message digest of its share:$(x_j, H(j))$, the merchant identity M and the payment order signed by M;

Other mobile agents can authenticate *Lagent* using the message digest. They verify the payment order using the merchant s public key. If anything inconsistent is found, the payment is stopped and the problem is reported to the owner.
2. Other $m \otimes 1$ Mobile Agents Send Shares to the Merchant
All the selected mobile agents send their shares of the secret key to the Merchant.

$$\text{Mobile agent} \otimes\!\!\rightarrow M : (Tid, Share_i, x_i)^{P_M}.$$

3. The Merchant Sends the Acknowledgement to *Lagent*
After $m \otimes 1$ shares have been received by the merchant, the merchant computes the message digests of the $m \otimes 1$ shares using the same hash function $H(\ )$. Then he sends the following message as a receipt to *Lagent*.

$$M \rightarrow Lagent\text{: } S_M(F_{REC}, Tid, M, Lagent, m \otimes 1 \text{ pairs of } (x_i, H(i)))$$

4. ***Lagent*** **Sends Its Share as the Last Share to** ***TTP***
Lagent veri es that each share received by the merchant is valid by comparing each message digest in the receipt with the one carried by itself. After that *Lagent* sends the mth (or last) share to the TTP.

$$Lagent \rightarrow TTP\text{: } (F_{SUB}, Tid, Lagent, M, e_s(C), Share_m, x_m)^{P_{TTP}}$$

5. ***M*** **Retrieves the Con rmed Message from** ***TTP***
Both *Lagent* and M have to retrieve the con rmed message from TTP as part of the non-repudiation evidence required in a dispute. It is assumed that even in the case of network failures, both parties will eventually be able to retrieve the message from TTP.

$$M \leftrightarrow TTP\text{: } (F_{CON}, Tid, Lagent, M, e_s(C), Share_m, x_m)^{V_{TTP}}$$
$$Lagent \leftrightarrow TTP\text{: } (F_{CON}, Tid, Lagent, M, Share_m, x_m)^{V_{TTP}}$$

6. Reconstruction of the Secret Key and Payment
Now, the merchant gets all the m shares, and all of them are real. He reconstructs the secret key s using the Lagrange interpolation formula.

$$s = a_0 = \sum_{j=1}^{m} Share_j \prod_{1 \quad k \quad m, k \neq j} \frac{x_k}{x_k \otimes x_j}.$$

The merchant decrypts the e-cash using $C = d_s(e_s(C))$, signs the e-cash with his private key V_M and forwards it to the bank.

5 Analysis of the Protocol

5.1 Correctness of the Protocol

First we show that the secure payment protocol (SPP) presented above provides protection against all the threats enumerated in the previous section. We do so through a series of simple claims.
Claim 1. The protocol SPP ensures that the e-cash is protected against spying/stealing by $m \otimes 1$ or less malicious MA hosts.
Proof. The e-cash is protected by encryption and it requires at least m agents to reconstruct the encryption key. Since at no point of the protocol any one agent has access to more than one share, this property is trivially guaranteed provided the agents are hosted by distinct hosts.

Notice that the protocol will fail if there is collusion between the merchant and the host of *Lagent*. Since, the merchant receives $m \otimes 1$ shares from the other $m \otimes 1$ agents, it simply needs to pass them to the host of the *Lagent*, who then uses that information to extract the e-cash.
Claim 2. The protocol produces evidence to support non-repudiation for both the customer and the merchant.
Proof. The non-repudiable evidences are generated at steps 3, 4 and 5 of the payment phase. At step 3, the merchant signs and sends message digests of the

$m \otimes 1$ shares already received by him. If these digests are not all valid, the *Lagent* will not complete the payment. If they are valid, they serve as evidence of M's receipt of the $m \otimes 1$ shares.

In step 4, the *Lagent* passes the last share as well as the encrypted e-cash to the TTP. This message is protected by encryption using the public key of the TTP. This ensures that the merchant cannot spy out the last share from this message. This message need not be signed by *Lagent* as TTP is trusted. (Otherwise it could have passed the message clandenstinely to the merchant, later corrupting the data and producing an untenable non-repudiation evidence).

In step 5, both M and *Lagent* retrieve the message using ftp, which serve as the non-repudiable evidence of transfer of the last share as well as the e-cash.

In summary, at the end of this protocol, if *Lagent* wants to prove that the shares have been received , she presents $S_M(F_{REC}, Tid, M, Lagent, m \otimes 1$ pairs of $(x_i, H(i)))$ and $(F_{CON}, Tid, Lagent, M, e_s(C), Share_m, x_m)^{V_{TTP}}$ to the judge. The first piece of evidence confirms that M received the $m \otimes 1$ shares and the second piece confirms that the last share was deposited with the TTP, which means that the merchant has access to it.

5.2 Efficiency of the Protocol

The message complexity of the protocol can be computed as follows. In the initialization phase, the *Lagent* sends $m \otimes 1$ messages of $O(1)$ length to $m \otimes 1$ participating agents. Each of the agents transfers its share to the merchant, using altogether $m \otimes 1$ messages of $O(1)$ length.

The merchant sends the single acknowledgement message of length $O(m \otimes 1)$ in step 3. In steps 4 and 5, three messages are transmitted, each of length $O(1)$. Thus altogether, the complexity of the messages communicated is $3O(m \otimes 1) + 3O(1)$, whereas the total number of messages transferred is $3(m \otimes 1) + 3 = 3m$.

The parameter m can be viewed as a measure of untrustworthiness of the host network. Therefore, it can be said that the cost of protection increases linearly with the number of untrustworthy hosts in the network.

It is worth noting here that the protocol involves the TTP only for the transfer of the last share. So the TTP remains unaffected by the number of shares m used for the payment.

6 Conclusion

The use of mobile agent technology is limited by the lack of a proper security framework. The security problem becomes even more serious when mobile agent is used in e-commerce systems in which processing and transfer of many sensitive information is involved. In this paper, we have addressed the problem of protecting sensitive information carried by mobile agents from malicious hosts, and proposed a payment protocol using Shamir's secret sharing scheme. The protocol guarantees protection of confidential data such as electronic cash against concerted attack by a known maximum number of malicious hosts. In addition, by using a TTP in a minimal way, it produces non-repudiable evidence of transfer of fund from the customer to the merchant.

The problem of protecting mobile agents against malicious hosts is acknowledged to be an extremely di cult one. While a general solution remains elusive,we have demonstrated here that speci c protections can be built using simple techniques. However, in order to provide a solution to be adopted in practice, more complex payment requirements as well as threat scenarios should be dealt with. For example, a hidden assumption for our protocol is that, the amount to be paid is pre-determined, which is obviously quite restrictive. A more general solution should address the case of arbitrary amounts of payment that are determined by the agents in the process of negotiation. Though the solution presented here can trivially be extended to cover the case of arbitrary amounts by using multiple e-coins, such a solution is certainly not elegant and may not be practical.

Another important issue is the scope of the malicious attacks against which protection must be provided in any practical secure payment protocol. The complexity of the solution is bound to increase as the threat model is expanded. One of the assumptions for this protocol is that there is no unholy alliance between the merchant and the bunch of crooked hosts. This assumption is unlikely to inspire con dence among the users of the payment system, and one of our next objectives is to nd an e cient way to relax this condition, with the minimal possible involvement of the TTP.

Ultimately how practical these solutions will be depends on how well these piecemeal approaches can be integrated into a compatible and e cient comprehensive solution against various possible threat scenario.

References

1. D. Chaum. Blind signature for untraceable payments. In *Proceedings of Crypto'82*, Plenum,NY, 1982.
2. D. M. Chess. Security issues in mobile code. *G.Vigna(Ed), Mobile agent and security , Lecture notes in Computer Science 1419, springer,Berlin*, 1998.
3. E. Dawson and D. Donovan. Shamir's scheme say it all. *Computer & Security(A-37) E.G.Dougall(Editor) Elsevier Science Publishers B.V.(North-Holland)*, 1993.
4. Y. Tsiounis G. Davida, Y. Frankel and M. Yung. Anonymity control in e-cash systems. In *Proc. Financial Cryptography Workshop*, pages 1–16, February 1997.
5. Fritz Hohl. Time limited blackbox security: Protecting mobile agents from malicious hosts. *Mobile agent and security , Lecture notes in Computer Science 1419, springer,Berlin*, 1998.
6. T. Sander and C. Tschudin. Towards mobile cryptography. Technical Report TR-97-049, International Computer Science Institute, Berkeley, November 1997.
7. A. Shamir. How to share a secret. *Communications of the ACM*, 22, November 1979.
8. V. Swarup W. Farmer, J. Guttmann. Security for mobile agents: Issues and requirements. In *Proceedings of the National Information Systems Security Conference*, 1996.
9. Digicash website. http://www.digicash.com.
10. J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *Proceedings of 1996 IEEE Symposium on Security and privacy*, Oakland, CA, May 1996.

Building Trust for E-Commerce: Collaborating Label Bureaus

Michael Shepherd, Anil Dhonde, and Carolyn Watters

Web Information Filtering Lab
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada B3H 1W5
{shepherd | dhonde | watters} @cs.dal.ca

Abstract. Consumers develop trust in a business through reports about that business from trusted third parties or other consumers. For the consumer to develop trust in e-commerce, a flexible system is required that rates e-commerce sites along multiple dimensions (such as delivery, return policies, etc.) and allows the consumer to determine which dimension(s) is of importance at that moment in time. These rating should be done by trusted third parties and stored in Label Bureaus. A prototype architecture has been developed that supports distributed Label Bureaus and multiple rating systems that rate along multiple dimensions. The architecture is based on a Label Engine and an Application Server that retrieves labels from one or more Label Bureaus and applies a rating algorithm to merge the various rating systems. An architecture that supports third party labeling allows the consumer to develop a rational trust level in unknown businesses.

1 Introduction

E-commerce on the Web is made up of two major categories of transactions; business-to-business (B2B) and business-to-consumer (B2C) [8]. While B2C transactions on the Web are growing, they are not growing as quickly as B2B transaction. Forrester Research expects that B2B e-commerce will be responsible for approximately $1.5 trillion in transactions by 2003, 14 times larger than their estimate for B2C transactions [10]. By this estimate, B2C would represent only about 7% of such transactions by the year 2003. Other, more conservative estimates, have the B2C share of e-commerce falling from its current 27% of dollar transactions to 17% by the year 2002 [15]

There are many reasons for the slower growth in B2C than in B2B e-commerce, but one reason has to do with the much slower development in e-commerce supporting services (such as security and payment services) on which B2C e-commerce relies. These supporting services are important in creating the "legitimacy" conditions required for trust to develop on the part of the consumer.

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 42-56, 2001.

Some important “legitimacy” conditions [15] that allow trust to develop include:

- Legitimacy conditions for consumer
  - the sellers are who they claim to be
  - the seller has right of sale over the item in question
  - the transaction and payment mechanisms are available, legal and secure
  - information about the buyer is not redistributed to other organizations or used for other purposes than for which it was intended
  - the item sold corresponds to its description and is suitable for its intended purpose
  - the purchased item can and will be delivered to the buyer

- Legitimacy conditions for the seller
  - the buyers are who they claim to be
  - the buyer has the resources to purchase the item

In this research, we have concentrated on the issue of trust from the perspective of the consumer. This is a difficult issue to deal with as it cannot be resolved by strictly algorithmic mechanisms. Current approaches include seals of approval and trust marks. The Better Business Bureau Online [3] has two programs, a privacy seal program and a reliability seal program. After evaluation by the BBBonline, a company can receive either or both seals that can be displayed on their Web site. TRUSTe [16] awards a seal or trustmark to an e-commerce site for display on their Web site if the company adheres to established privacy principles such as indicating to the user what information is being gathered and how it will be used, etc. Such seals are valid, of course, only at the time of the evaluation itself.

A major problem with seals is that a seal represents a summarization of a number of different dimensions along which that business has been evaluated and the consumer may not be aware of the dimensions, the scale of allowable values on each dimension, or the values that were assigned to individual dimensions. The consumer only sees that the business in question either has the seal or that it does not have the seal. At best, the seal may have different levels, but this is still a rating on a single dimension of approval. To interpret the seal, the consumer must check with the issuing authority to determine the date of issue and what the level means. An alternative approach used by eBay [4] is to gather feedback from both buyers and sellers on each other. Comments are gathered and buyers and sellers are awarded one positive point for each positive comment and a negative point for each negative comment. Both the points accumulated and the actual comments are available to everyone. Although it is an interesting and useful evaluation, it is not controlled and the evaluators are not necessarily trusted third-parties.

A better approach would be the use of labels where a label consists of a set of dimensions and the business has been rated and assigned a value on each of these dimensions. For example, an e-commerce Web site might be rated by a rating service on the dimensions of security of communications, timely delivery of goods ordered and having a dispute resolution mechanism. To engender trust, such ratings must be done by trusted third parties and stored in independent label bureaus, as described below. Then, the consumer can decide if they wish to do business with the target business not based on an overall evaluation of that business but by an evaluation of

the scores along those dimensions that are of importance to that consumer for that transaction.

For example, a consumer may need a certain item immediately. Business A is rated poor for its time-to-delivery of purchased items but very high on security and privacy of both communications and the Web site itself. Business B is rated excellent on its time-to-delivery but poor on security. The choice is now the consumer's; how badly does the consumer need that item and how important is the issue of security to that consumer in this particular instance?

A problem arises, of course, if the label bureau has a label for one but not the other of these businesses. Also, the consumer may want a second opinion, i.e., the consumer may not be comfortable with a label from a single label bureau. In this case, multiple label bureaus would have to be consulted and their labels integrated in some fashion. A similar system is being developed for the rating of medical information [6]. In that scenario, doctors, medical societies and associations, will critically appraise Internet information and act as third-party raters of the information with respect to the value and trustworthiness of the information. Consumers and professionals may subscribe to a multitude of these services to get, automatically, ratings from different perspectives when retrieving information from the Internet. Khare and Rifkin [7] have also suggested that security metadata can be stored in label bureaus and these labels may have multiple dimensions, such as, principals' clearance levels and capabilities required for action. In order to make policy decisions, the opinions of several ratings in different systems may be weighted and considered

The aim of this research is not the development of rating systems for e-commerce, rather it is the development of a generic architecture for collaborative label bureaus as described above. The remainder of this paper describes label bureaus and rating systems in general and how they can be used for building trust in B2C transactions. The Label Bureaus and labels are based on the Platform for Internet Content Selection (PICS™) [9] and the associated specification for PICSRules [5]. These concepts are then expanded to describe the generic architecture that allows collaborative rating based on the combination of labels from independent label bureaus.

2 Label Bureaus

Label Bureaus were developed initially as trusted third-parties for the storage and distribution of labels to be used by Web browsers and applications to protect children, as much as possible, from *unintentional access* to possibly legal, but nonetheless objectionable material [14].

Labeling refers to schemes to assign content related labels to URL's and/or specific Web pages. The individual rating protocols exist, in general, separate from products or applications using these ratings. These labels can be stored as part of the Web page or separately from the Web page in a database.

Labels may be the result of [1]:

- Self-rating (first-party). Self-rating refers to the practice of Web page creators to provide a label describing the content of those pages.

- Community rating by interested users (second-party). Community rating schemes are informal efforts by like-minded consumers of Web pages to share their knowledge of the content of Web pages with each other.
- Third-party authority rating. Community rating schemes are informal efforts by like-minded consumers of Web pages to share their knowledge of the content of Web pages with each other.

The quality of labels produced by first-party and second-party rating reflects the self-interest of the producer, whereas labels produced by third-party rating should be more independent.

In this paper, we restrict our view to trusted third-party raters and the independent Label Bureaus in which these labels are stored. Figure 1 is an example of label bureaus and their use. The third-party raters examine a Web site and generate a label and store it in the Label Bureau. When users wish to access the Web site, the labels for that site are retrieved from the Label Bureau and may be used in making decisions regarding that site.

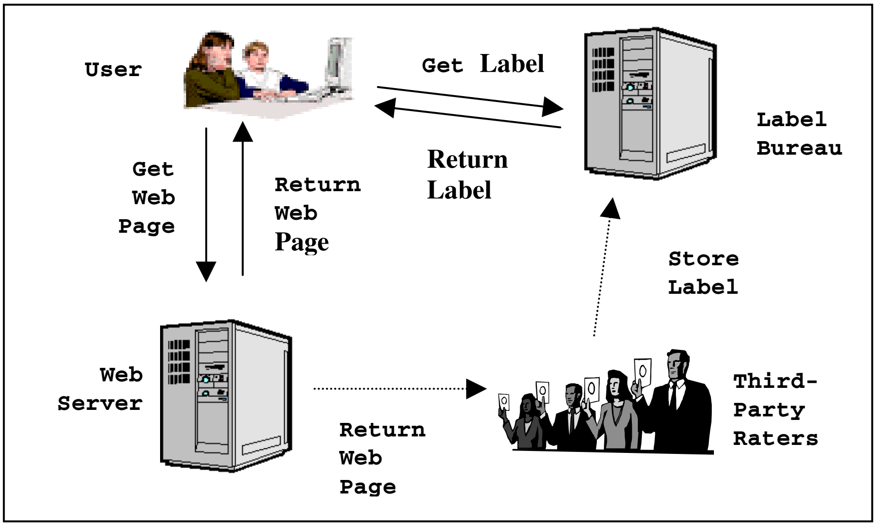

Fig. 1. Label Bureau Access

Although different labeling protocols or schema are available, only a few are in wide use on the Internet and currently deal with the content of the Web site rather than the rating of the business behind the Web site. Nonetheless, the structure of these schema can used in the e-commerce domain. Two examples of schema for Web filtering for children are:

- The RSACi system [12] rates sites on four cateories (language, nudity, sex, and violence) and provides users with information on the levels (0 to 4) of these

categories in given sites. RSACi is now used by Internet Explorer, CyberPatrol, and CompuServe (in the USA and in Europe).

- SafeSurf [13] is an eleven category rating scheme (age range, profanity, heterosexual themes, homosexual themes, nudity, violence, (sex, violence, and profanity), intolerance, glorifying drug use, other adult themes, gambling). Each major category is subdivided into multiple subcategories. SafeSurf is used by both Netscape Navigator and Internet Explorer.

A layer cake model [1] for the combination of first and third-party ratings schemes has been proposed as part of the Self Regulation of Internet Content project [2], and has relevance to B2C e-commerce. As shown in Figure 2, such a model has three layers, all resting on a "plate" of technologies such as PICS and XML for implementation of the model. The first layer of the cake is a basic vocabulary that will be used by first parties in rating their sites. The second layer of the cake consists of ratings templates created by third parties. These templates may reflect different legal systems, cultures, or interest groups. The third layer of the cake consists of a set of ratings of individual sites created by trusted third parties using the vocabularies and templates of levels one and two. This level also includes filtering software to be applied to these ratings. By separating the vocabulary elements from the construction of templates, we can better allow third parties to reflect their value systems while still preserving the vocabulary of the first-party raters.

3 Platform for Internet Content Selection (PICS)

The PICS specification enables labels (metadata) to be associated with Internet content [17]. It was originally designed to help parents and teachers control what children access on the Internet, but it also can be used as a platform on which other rating services and filtering software can be built. In this research, PICS labels are used to rate e-commerce sites along multiple dimensions in an effort to create the legitimacy conditions for the development of trust on the part of the user.

The use of labels for site description requires several steps: creation of labels, storage of labels, maintenance of labels, dissemination of labels, authentication of label values, and finally the use of available labels by the filtering software to classify given entities as acceptable or not acceptable for the given context. The primary platform for associating labels with Web pages and specifying the format of these labels is PICS. PICSRules [5] form the primary platform for writing profiles for the filtering of Web pages based on these labels.

PICS specifies three methods by which these labels may be transmitted: in an HTML document using the META tag, with a document transported via a protocol that uses RFC-822 headers, and separately from the document where a client requests labels from a Label Bureau that runs the HTTP protocol. In our architecture, only this third method , i.e., requesting the label from a Label Bureau, is of interest.

The following is an example of a PICS label list, adapted from the W3C example, as it might be applied in an e-commerce situation. In this example,

"http://www.ecomm-rating-service.org/v1.5" is the URL of the rating service that produced the label for the company represented by the site at http://www.xyz-ecomm.com/. The label was created by John Doe and is valid from November 5, 2000 until November 5, 2001. The label is providing ratings along three dimensions. Assuming that the ratings are on the scale of 0 through 10 where 0 is poor and 10 is excellent, then the rating for this e-commerce site was poor for security of payments and customer data, but was very good for fast delivery.

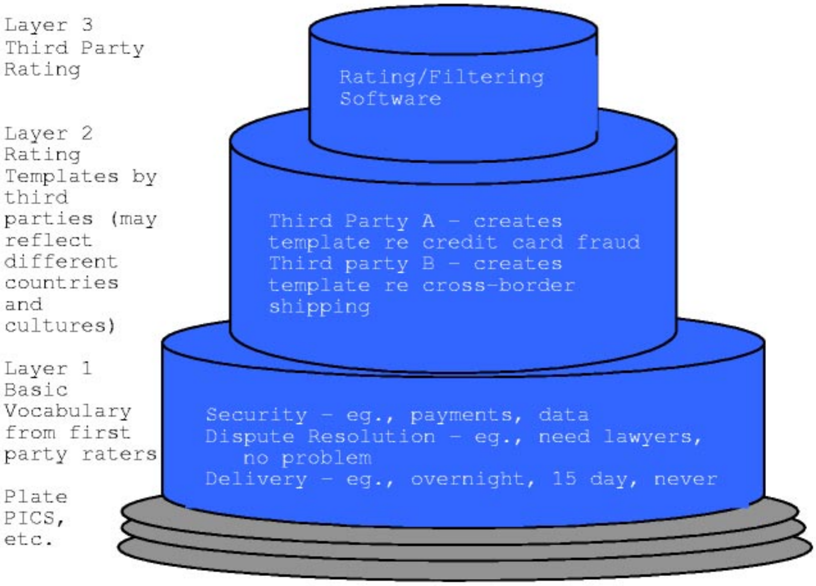

Fig. 2. Layer cake model

```
(PICS1-1 "http://www.ecomm-rating-service.org/v1.5"
   by "John Doe"
   labels on "2000.11.05"
   until "2001.11.05"
   for http://www.xyz-ecomm.com/
   ratings(secure_payment 2 secure_data 2 fast_deliv 8)
)
```

The following example PICSRule could be applied to this label. The rule would alert the consumer before the transaction takes place if the company associated with this label is either rated very poorly with respect to secure payments or not rated highly enough for its ability to delivery in a timely fashion. The attribute "secure_data" is not pertinent to this rule and is ignored.

```
(PicsRule-1.1
   (
   serviceinfo (
      http://www.ecomm-rating-service.org/v1.5
```

```
        shortname "ECR"
        bureauURL http://www.ecomm-label-bureau.org
                  /ratings
        UseEmbedded "N"
        )
    Policy (RejectIf "((ECR.secure_payment < 2) or
                      (ECR.fast_deliv < 7))")
    Policy (AcceptIf "otherwise")
    )
)
```

In this example, labels embedded in documents are ignored (UseEmbedded "N") and only labels retrieved from the Label Bureau site (http://www.ecomm-label-bureau.org /ratings) using the rating scheme "http://www.ecomm-rating-service.org/v1.5" are used to assess the companies. The rating scheme specifies the dimensions used for labeling, the scale of allowable values on each dimension, and a descriptor of the criteria used in assigning values [11].

The policy statements are the actual rules to be applied to this label. The policy is to alert the consumer if the company is rated as less than 2 for secure payment or less than 7 for fast delivery. In this instance, the transaction can proceed. The policy or rule ignores the "secure_data" attribute of this label.

4 Architecture

It was felt that, to build trust, it would be best to have multiple third-party ratings of individual companies. The third-party resolves the issue of self-interest found in first-party rating. Integrating multiple third-party ratings would give some protection against both overly positive and overly negative evaluations. In the architecture that has been developed, these multiple ratings are stored in multiple Label Bureaus and a single Label Engine maintains a database of Label Bureaus. Multiple labels are integrated by the Application Server. Thus, a client request for labels goes to the Label Engine, not to the individual Label Bureaus. There may be multiple Label Engines distributed over the Web or there may be a single Label Engine to service an enterprise. In our prototype, we implemented a single Label Engine only.

This architecture provides the user with the following capabilities:

- **Rating**. Labels can be retrieved from various Label Bureaus regarding an e-commerce site. The algorithm described below combines the labels from these Label Bureaus into an overall rating system for the consumer.
- **Filtering**. PICSRules are applied to this overall rating to determine if the e-commerce site meets the consumer's criteria.
- **Ranking**. Those companies that pass the filter can be presented in a rank order dependent on their derived ratings.

The architecture of the system is shown in Figure 3. The third-party labels are stored in Label Bureaus distributed over the Internet. The Label Engine maintains a data base of these Label Bureaus. Our prototype system was implemented in Java using servlets to maintain state over the course of a session. The data at the Label Engine and the Label Bureaus were stored in the Oracle DBM Systems and accessed via JDBC.

In processing a client request, the following steps take place:

1. The Application Server receives a request from a client machine for information about an e-commerce Web site,
2. The Application Server issues a request to the Label Engine for the list of Label Bureaus that hold labels rating the e-commerce of interest,
3. The Label Engine issues a request to all of the Label Bureaus in its data base asking if they hold labels for this e-commerce site,
4. The Label Bureaus return a simple yes/no to the Label Engine,
5. The Label Engine then returns the URLs of those Label Bureaus holding such labels to the Application Server,
6. The Application Server then makes requests for those labels directly to the multiple Label Bureaus and requests the associated rating systems from the Rating Services
7. Upon receiving the labels and the rating systems, the Application Server combines these labels and their associated rating systems to produce an overall rating for the e-commerce site and, possibly, to filter out the site.

5 PICSRules in Context of This Architecture

PICSRules, although commonly used for safe Internet browsing and content filtering, can be used to described e-commerce site and services. In this architecture, PICSRules are stored at the Application Server. Although we supplied the rules, it would be fairly straight forward (although perhaps not advisable for naïve users) for the users to create rules through a GUI.

In this system, the PICSRules are used for the following purposes:

- List Rating Services and Label Bureaus
- Set user preferences for computing the ratings. Select whether to rate, filter or rank the site(s)
- Select the rating algorithm, if the server allows multiple rating algorithms for computing the combined rating.
- Select a specific dimension, if the user decides to get the rating on that particular category.
- Optionally set the weights of the dimensions, if the user is not satisfied with the default weights.

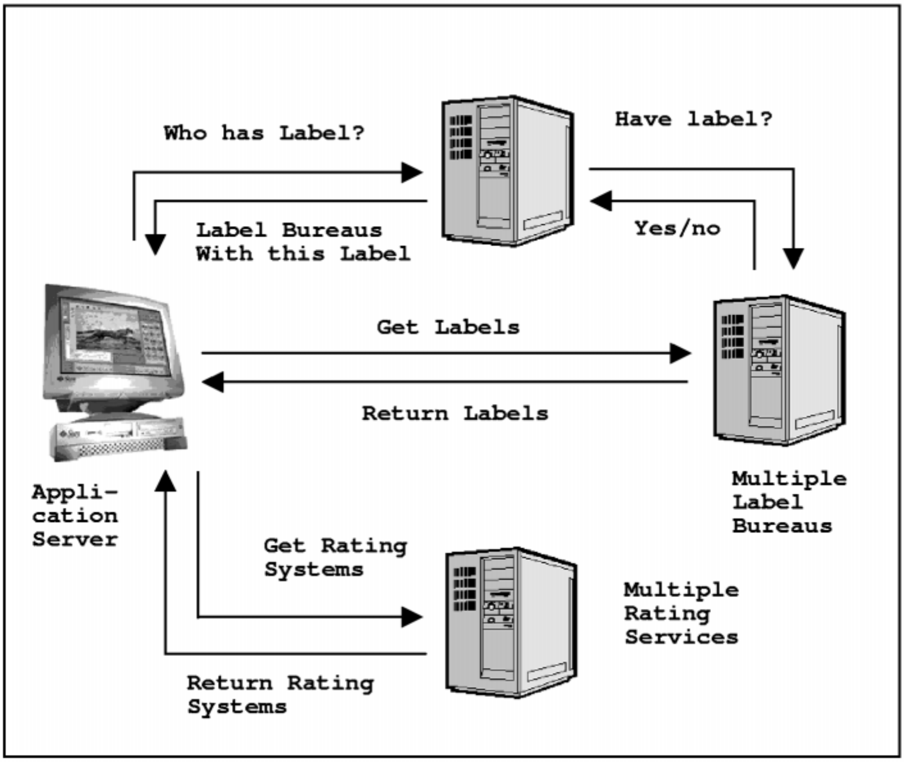

Fig. 3. Architecture

It would be easier to explain and understand the above mentioned features of the PICSRules with an example from our prototype system. In the example shown below, only labels retrieved from the Label Bureau site "http://borg.cs.dal.ca:2084" using the rating scheme "http://borg.cs.dal.ca:2084/1.rat" are used to assess the e-commerce site. This is specified in the *ServiceInfo* portion of the rule.

ServiceInfo is used to specify a user selected Label Bureau and Rating Scheme for use in the evaluation. In such a case, even if the Label Engine does not include this Label Bureau in its list, the rating process will access their labels. So the labels from this Label Bureau will be procured along with the Label Engine specified Label Bureaus.

This rule also has two required extensions (*reqextension*). These extensions permit the consumer to specify certain options (*ratOption*) and to specify the Label Engine to the Application Server (*engineExtension*). These must be included in the rating process. If there is any error while retrieving these extensions, then the rating process fails, i.e., returns an error to the user.

ratOption is a sample extension that allows the user to chose one of the three options: rating, filtering or ranking. This is declared in the option dimension of the extension. Here the specifyBureau is the user specified Label Bureau to be included in the evaluation. The user has also specified the Label Engine to be used in this rule,

hence the specifyEngine dimension. However the user wants the Application Server to use its default rating algorithm.
engineExtension is an extension created to specify the Label Engine details such as address and port number. If the Label Engine is unavailable, the rating process should terminate and return an error. This is specified by the UnAvailable option. The user desires to use only one engine to obtain the rating; hence the specified multipleEngine option as “NO”.

```
(PicsRule-1.1
    (
        ServiceInfo(
                    "http://borg.cs.dal.ca:2084/1.rat"
                     shortname "eRat"
                     bureauURL"http://borg.cs.dal.ca:2084
"
                    )
        reqextension(

"http://borg.cs.dal.ca:2086/ratOption.html"
                     shortname "ratOption"
                      )
        ratOption(
                  option "RATING"
                  specifyBureau "YES"
                  specifyEngine "YES"
                  ratAlgorithm "DEFAULT"
                   )
        reqextension(

"http://borg.cs.dal.ca:2085/engineExtension.html"
                     shortname "engineEntension"
                      )
        engineExtension(
                        address "http://borg.cs.dal.ca"
                        port "2081"
                        UnAvaliable "FAIL"
                        MultipleEngine "NO"
                        )
   )
)
```

6 The Default Rating Algorithm

The Application Server is responsible for applying the rating algorithm to the collected labels and rating systems. The rating algorithm provides the logic for processing labels in order to compute a rating value for the company associated with

the given URL. The architecture is not based on a specific rating algorithm, but is designed in such a way that any rating algorithm can be incorporated.

The default rating algorithm permits the combination of different rating systems, with the user specifying the weight to be assigned to the values in any given dimension of a label. For example, in Figure 4, the default rating algorithm is applied to determine an overall evaluation of a particular e-commerce site. In this example, there are labels from two rating systems, A and B, to be combined. The dimension names and values are given in the boxes, with the security dimension of rating system A and the dispute resolution dimension of rating system B having sub-dimensions. All dimension weights are on the 0-10 scale with 10 being good. The user has assigned proportionate weights to each dimension, as shown in parentheses.

Weighted averages are calculated from the leaves to the root of the tree to determine the final result. In Figure 4, the value of each leaf node is assigned as shown and the value of each interior node is calculated as follows:

- value of the security dimension of scheme A is calculated as (5*.6 + 7*.4) = 5.8
- value of rating system A is calculated as (5.8*.8 + 6*.2) = 5.84
- value at dispute resolution dimension of scheme B is calculated as
- (6*.7 + 3*.3) = 5.1
- value of rating system B is calculated as (6*.6 + 5.1*.4) = 5.64
- rating systems A and B contribute equal amounts to the final result value of for this e-commerce site and calculated as: (5.84*.5 + 5.64*.5) = 5.74

Further PICSRules would then be applied in order to determine whether or not to accept or to filter out this e-commerce site.

In order to allow the user to specify the rating algorithm, we make use of the extension clause of PICSRules. In the above PICSRule example, this extension is termed as “ratOption” with the dimension ratAlgorithm specifying the name of the rating algorithm. In the above example, a default ranking algorithm was applied.

However, if the user wishes to compute the rating on the basis of a particular dimension, the user can mention it in the PICSRules itself, using the extension clause termed “dimensionExtension” to specify the dimension. The nomenclature of extension is left for individual implementations.

If the dimension is mentioned, then only labels for those Rating Systems having that dimension are requested and the rating is calculated for that particular dimension only. For example in our rating algorithm, we would bubble the values up the dimension tree only till that dimension is reached, then use this rating as final for that dimension tree. If multiple labels are available for that dimension, the rating algorithm treats these dimension sub-trees as rating trees and combines their ratings.

These user-specified weights can be specified in an extension clause, which is interpreted by the Application Server. It is important to note that these weights are used in our rating algorithm, which is not a mandatory part of the architecture. Software has also been develop for creating the PICRules and labels and for the retrieval and viewing of the e-commerce sites, including the labels and rules.

Figure 5 is a sample of the end-user interface. The Web page has three frames. The main frame is used to display the target Web page. In Figure 5, this frame displays the eBay home page.

The top frame allows the user to input the URL of a target Web site and the three buttons: Display Web Site, Get Rating, Filter <on/off>. This system produces a rating of a Web site as determined by the associated labels and PICSRules. This rating may be used to filter the Web site if the filter option is "on". The user may also request a display of the rating and how it was obtained. If the default rating algorithm was used then the weights at all the leaf nodes and the derived weights at the interior nodes are shown.

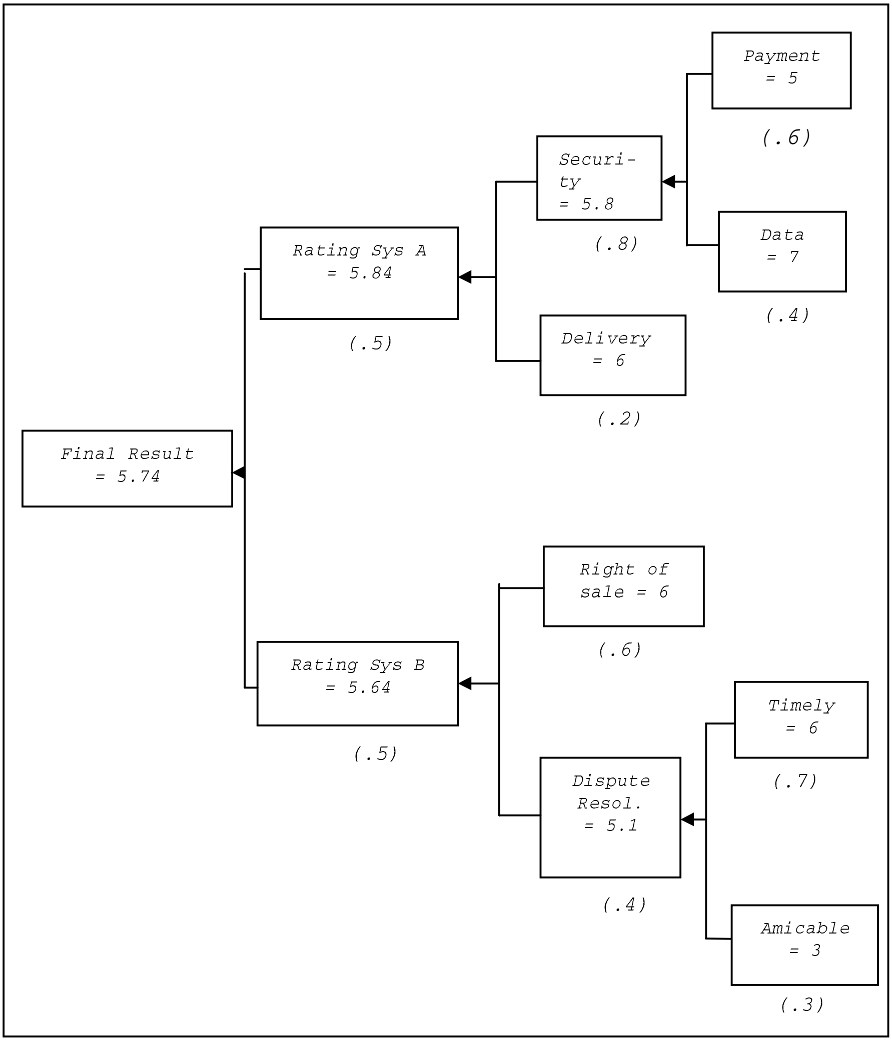

Fig. 4. Default rating algorithm

The left frame provides the options of reviewing information pertaining to the rules and labels, such as, viewing the PICSRules being used for this user and viewing the label hierarchy and rating system for the labels associated with this Web site.

Fig. 5. User interface

7 Summary

In summary, we have developed a prototype architecture that permits distributed Label Bureaus and multiple rating systems to be combined in such a fashion that users can rate, filter and rank e-commerce sites on the Web. PICS and PICSRules are used as the platform for the implementation. The rating can be done either by specific category or by a default algorithm that considers all dimensions. The architecture is sufficiently flexible to permit various rating algorithms to be applied.

From the perspective of the customer, the "legitimacy" conditions that are required for the development of trust for B2C e-commerce include:

- the sellers are who they claim to be
- the seller has right of sale over the item in question
- the transaction and payment mechanisms are available, legal and secure

- information about the buyer is not redistributed to other organizations or used for other purposes than for which it was intended
- the item sold corresponds to its description and is suitable for its intended purpose
- the purchased item can and will be delivered to the buyer

The prototype system, as briefly described in this paper, is an e-commerce support service that can help create these legitimacy conditions. PICS labels supports multi-dimensional rating schemes, thus permitting e-commerce sites to be rated along multiple dimensions and allowing consumers to have access to these ratings. These dimensions may include all of the above legitimacy conditions and may be arranged in a hierarchical manner. The layer cake model permits different third-party templates to be applied, allowing for differences in countries and cultures, or even such simple differences as whether a scale is binary (has right to sell or does not have right to sell) or non-binary (we are somewhat confident they have the right to sell). Most importantly, consumers can generate their own PICSRules specifying how they wish labels to be processed. This includes the ability to combine labels generated by different rating services using different rating schema.

It is proposed that rating systems be developed along appropriate dimensions, that trusted third-party rating be done and that the resulting labels be stored in Label Bureaus. If such labeling is done, then a distributed network of Label Bureaus with multiple rating schemes and some architecture, such as we have prototyped, can be used to access and use this information effectively.

References

1. Balkin, J.M., Noveck, B.S., Roosevelt, K.: Filtering the internet: A best practices model. Information and Society Project. (1999). Available: http://www.law.yale.edu/infosociety/filtering_report.html [2000, December]
2. Bertelsmann Foundation. Self Regulation of Internet Content. Available: http://www.stiftung.bertelsmann.de/internetcontent/english/frameset_home.htm [2000, December]
3. Better Business Bureau (Online). Available: http://www.BBBonline.org [2000, December]
4. eBay. Available: http://www.eBay.com [2000, December]
5. Evans, C., Feather, C.D.W., Hopmann, A., Presler-Marshall, M., Resnick, P.: PICSRules 1.1, W3C Recommendation (29 Dec 1997) Available: http://www.w3.org/TR/REC-PICSRules [2000, December]
6. Eysenback, G.: Collaboration for Critical Appraisal of Medical Information on the Internet. (1999) Available: http://www.dermis.net/medpics/ [2000, December]
7. Khare, R., Rifkin, A.: Trust Management on the World Wide Web. Available: http://www.cs.caltech.edu/~adam/papers/www/trust-management.html [2001, January]
8. Kinney, Sam: An Overview of B2B and Purchasing Technology. Response to Call for Submissions, Federal Trade Commission, Public Workshop: Competition Policy In the World of B2B Electronic Marketplaces. (2000) Available: http://mba.tuck.Dartmouth.edu/Anderson/Kinney_white_paper.htm [2000, December]

9. Krauskopf, T., Miller, J., Resnick, P., Treese, W: PICS Label Distribution Label Syntax and Communication Protocols, Version 1.1, W3C Recommendation (31-October-96). Available: http://www.w3.org/TR/REC-PICS-labels [2000, December]
10. Matthews, Jayson: Study Quantifies B2C Problems. *InternetNews*. (August 1, 2000). Available: http://www.internetnews.com/bus-news/articles/0,,3_427751,00.html. [2000, December]
11. Miller, J., Resnick, P., Singer, D.: Rating Services and Rating Systems (and Their Machine Readable Descriptions), Version 1.1, W3C Recommendations (31 Oct 1996). Available: http://www.w3c.org./TR/REC-PICS-services-961031 [2000, December]
12. RSACi Recreational Software Advisory Council.: (2000). Available: http://www.icra.org [2000, January]
13. SafeSurf: Available: http://www.safesurf.com [2000, January]
14. Shepherd, M., Watters, C.: Content Filtering Technologies and Internet Service Providers: Enabling User Choice, Industry Canada. Canada. March, (2000). Available: http://strategis.ic.gc.ca/internet [2000, December]
15. Tee, Koh Ai: E-Commerce in an Era of Creative Destruction. (2000). Available: http://www.alumni.nus.edu.sg/Alumnus/jul2000/ecom.html [2000, December]
16. TRUSTe. Available: http://www.Truste.com [2000, December]
17. W3C. Platform for Internet Content Selection (PICS). Available: http://www.w3.org/PICS/ [2001, January]

Group-Oriented (t, n) Threshold Digital Signature Schemes with Traceable Signers*

Zi-Chen Li[1,2], Jun-Mei Zhang[2], Jun Luo[2], William Song[3], and Yi-Qi Dai[1]

[1]Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, P.R.China
lzc@theory.cs.tsinghua.edu.cn
[2]Department of Computer Science and technology
Jiaozuo Institute of technology, Jiaozuo, 454159, Henan Province, P.R.China
[3]E-Business Technology Institute
Hong Kong University, Pokfulam Road, Hong Kong
wsong@eti.hku.hk

Abstract. In this paper, we propose two group-oriented (t,n) threshold digital signature schemes with traceable signers. One needs the assistance of a mutually trusted center, the other does not need the assistance of a mutually trusted center. We also discuss the security and the feature of both schemes and show that the new schemes can withstand conspiracy attacks without attaching a secret number and can avoid to the forgery attacks. In addition, we show that, by the new schemes, any outsider can trace back to find who the signers are.

Keywords: Information security, Threshold digital signature, Traceable

1 Introduction

Along with the swift and violent development of telecommunication and computer network, especially the popularization of the Internet, the electronic commerce systems based on the Internet came out and have flourished in the past few years. However, the problem of the security holds back its growing-up. The key technique in the security of electronic commerce is the digital signature of the transfer message and user identity authentication. Multi-singer digital signature and batch verifying protocol can increase the security and improve the efficiency of the digital signature scheme. According to the different properties, the multi-singer digital signature schemes can be classified into many types. Group signature and threshold signature is a major and important form of the multi-signer signature.

In 1991, Desmedt and Frankel [1] first proposed the concept of a group-oriented (t,n) threshold digital signature scheme based on RSA system [2]. A (t,n) threshold digital signature has the following properties:

* Supported by National 973 Project(No:G1998030420)
* Supported by the Fund of Henan Province Nature Science(No:004070400)

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 57-69, 2001.

(1) Any t or more group members can generate group signature on behalf of the group;
(2) Any less than t members cannot generate the legitimate group signature;
(3) Any outsider can use a group public key to verify the group signature;

Generally speaking, the group-oriented (t,n) threshold signature schemes are divided into two kinds:

(1) Group-oriented (t,n) threshold signature scheme with traceable signers,

(2) Group-oriented threshold signature scheme with anonymous signer

In addition to the above properties, in this first kind scheme, we can trace back to find the signers without revealing the secret keys, but in the second scheme, we cannot find the signers.

In the scheme designed by Desmedt and Frankel [1], the group secret key and the group member secret keys are determined by a trusted key authentication center. Harn [3] used the cryptographic technique of Shamir's perfect secret sharing scheme [4], which is based on the Lagrange interpolating polynomial and digital signature scheme to construct two group- oriented (t,n) threshold digital signature schemes. One requires the mutually trusted center and the other does not require the mutually trusted center. Unfortunately, the above schemes [1,3] suffer from the conspiracy attacks and the secret keys can be revealed with high probability [5]. To avoid the attacks and to be able to trace back to find the singers without the revealing the secret keys, Wang et al. [6] for the first time proposed two group-oriented (t,n) threshold digital signature schemes, one needs the assistance of a mutually trusted center, and the other does not need. However, Tseng and Jan [7] show that Wang et al. group-oriented (t,n) threshold digital signature schemes are insecure under a forgery attack. Li et al. [8, 9, 10] also show that any malicious attacker without any secret keys of group and group members can not only forge a valid group signature for any message, but also forge valid individual signature.

In this paper, we will propose two group-oriented (t,n) threshold digital signature schemes with traceable signers. Both schemes can withstand conspiracy attacks without attaching a secret number as in the reference [5] and can avoid to the forgery attacks proposed in [7, 8]. One needs the assistance of a mutually trusted center and the other does not need the assistance of a mutually trusted center.

In the next section, we will propose modified ElGamal digital signature scheme, which is developed from generalized ElGamal type digital signature schemes [11]. In the section 3 and 4, we will present two new group-oriented (t,n) threshold digital signature schemes with traceable signers, one needs a mutually trusted center and the other does not need the trusted center. Some features and security of two new group-oriented (t,n) threshold digital signature schemes are discussed and analyzed in the section 5. Finally, we conclude the paper in the last section.

2 Modified ElGamal Digital Signature Scheme

In this section, we will propose modified ElGamal digital signature scheme, which is developed from generalized ElGamal type digital signature schemes [11].

System parameters:

p : a large prime number, $p = 2p \ \ +1$, p is also a large prime number;

g : a primitive element of $GF(p)$;

H : a one-way function;

x_U : a secret key of user U ;

y_U : a related public key of user U, $y_U = g^{x_U} \bmod p$.

Signature generation:

Assume that user U wants to sign a message m. User U randomly selects a number $k \quad [1, p \quad 1]$, and computes

$$r = g^k \bmod p. \tag{1}$$

User U soles the following congruence for integer s, where $m \ = H(m)$.

$$rs = (r + m \ )k + x_U \bmod p \quad 1 \tag{2}$$

The signature for message m is then the ordered pair (r, s).

Signature verification:

Upon receiving the set of $(m; (r, s))$, any user can verify the signature of message m as

$$g^{rs} = r^{r+m} y \bmod p \text{, Where } m \ = H(m). \tag{3}$$

By the way, the security analysis of the new modified scheme is very similar to the security analysis of the scheme proposed by Agnew et al. [12]. We will not discuss it here.

3 Group-Oriented (t, n) Threshold Digital Signature Schemes with the Assistance of a Mutually Trusted Center

In this section, based on the above new modified ElGamal's signature, we will propose new group-oriented (t, n) threshold digital signature scheme with traceable signers, in which the mutually trusted center is used.

System parameters:

The trusted center is responsible for selecting all parameters, the group secret key and group member secret keys.

p : a large prime number;

q : a large prime divisor of $p - 1$;

a_i: a random integer with $0 < a_i < q$, where $i = 0,1,\cdots,t-1$, $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} \bmod q$;

g : a generator with order q of $GF(p)$;

H : a one-way function;

The parameters $\{p, q, g, H\}$ are the public values, but $\{a_i, i = 0,1,\cdots,t-1\}$ are the secret values.

Group and group member secret and public key:

Each group member U_i ($i = 1,\cdots,n$) has the following secret and public keys computed by trusted center.

$f(ID_i) \bmod q$: a secret key of user U_i with public identity ID_i;

$y_i = g^{f(ID_i)} \bmod p$: a related public key of user U_i.

In addition, the trusted center computes group secret and public key as follow.

$f(0) \bmod q$: group secret key;

$y = g^{f(0)} \bmod p$: group public.

Individual signature generation:

This new scheme allows any t group members to represent the group to sign a message m. Without losing generality, t group members $U_1, U_2, \cdots U_t$ represent the group and sign a message m. The t group members can sign the message simultaneously.

Group member U_i randomly selects an integer, $k_i \in [1, q-1]$, computes $r_i = g^{k_i} \bmod p$, and make r_i publicly available through a broadcast channel. After all values are available, the t group members together computes the value.

$$R = \prod_{i=1}^{t} r_i \bmod p \tag{4}$$

Group member U_i uses his secret key, $f(ID_i)$ and k_i solves the following congruence equation for integer s_i, where $m' = H(m)$.

$$s_i R = (R + m')k_i + f(ID_i)\Big(\prod_{j=1, j \neq i}^{t} \frac{-ID_j}{ID_i - ID_j}\Big) \bmod q \quad (i = 1,\cdots,t) \tag{5}$$

Then, the set (s_i, r_i) is the signature for message m. Group member U_i transmits the signature to a designated clerk.

Individual signature verification:

On receiving the signature message (s_i, r_i) for message m from U_i, the clerk utilizes the public key y_i to authenticate the validity of the individual signature.

$$g^{s_i R} = r_i^{R+m'} y_i^{(\prod_{j=1, j \ne i}^{t} \frac{ID_j}{ID_i - ID_j})} \bmod p \tag{6}$$

If the above equation holds, the individual signature (s_i, r_i) from U_i is valid; otherwise, the signature is invalid.

Further, the clerk uses t pairs of public values (ID_i, y_i) to construct a Lagrange polynomial function $h(y)$ as following.

$$h(y) = \sum_{i=1}^{t} ID_i \prod_{j=1, j \ne i}^{t} \frac{y - y_j}{y_i - y_j}. \tag{7}$$

By the Lagrange polynomial function $h(y)$, we can use group member's public values to find the singer who signed the signature for message m.

Group-oriented (t, n) ***threshold signature generation:***

After t individual signatures are received and verified by the clerk, the group signature of the message m can be obtained as (R, S), where

$$R = \prod_{i=1}^{t} r_i \bmod p \tag{8}$$

$$S = \sum_{i=1}^{t} s_i \bmod q. \tag{9}$$

Group-oriented (t, n) ***threshold signature verification:***

Any outsider can use the group public key y to authenticate the validity of the group signature (R, S) for message m.

$$g^{SR} = R^{R+m'} y \bmod p, \text{ where } m' = H(m). \tag{10}$$

If the above equation holds, the group signature (R, S) is valid; otherwise, the group signature is invalid.

In fact:

$$s_i R = (R + m')k_i + f(ID_i)(\prod_{j=1, j \ne i}^{t} \frac{ID_j}{ID_i - ID_j}) \bmod q \quad (i = 1, \cdots, t) \tag{11}$$

$$\sum_{i=1}^{t} s_i R = (R + m') \sum_{i=1}^{t} k_i + \sum_{i=1}^{t} f(ID_i)(\prod_{j=1, j \ne i}^{t} \frac{ID_j}{ID_i - ID_j}) \bmod q \tag{12}$$

$$SR = (R+m)\sum_{i=1}^{t} k_i + f(0) \bmod q \text{, where } S = d\sum_{i=1}^{t} s_i \bmod q \tag{13}$$

$$g^{SR} = g^{(R+m)\sum_{i=1}^{t} k_i} g^{f(0)} \bmod p = R^{R+m} y \bmod p \tag{14}$$

Signer identity verification:
If the outsider wants to determine the signer of the group signature (R, S) for message m, he can use group member U_i's public values (ID_i, y_i) to authenticate the following equation.

$$ID_i = h(y_i)$$

If the above equation holes, the group member with public values (ID_i, y_i) is a signer of the group signature (R, S) for message m; otherwise she is not.

4 Group-Oriented (t, n) Threshold Digital Signature Schemes without the Assistance of a Mutually Trusted Center

In this section, based on the above new modified ElGamal's signature, we will propose another new group-oriented (t, n) threshold digital signature scheme with traceable signers, in which the mutually trusted center is no longer used.

System parameters:
There are some public parameters that should be agreed by all group members.

p : a large prime number;

q : a large prime divisor of $p-1$;

a_i: is a random integer with $0 < a_i < q$, where $i = 0,1,\cdots,t-1$, $f(x) = a_0 + a_1x + \cdots + a_{t-1}x^{t-1} \bmod q$;

g : a generator with order q of $GF(p)$;

H : a one-way function;

The parameters $\{p, q, g, H\}$ are the public values, but $\{a_i, i = 0,1,\cdots,t-1\}$ are the secret values.

Group and group member secret and public key:
Each group member U_i ($i = 1,\cdots,n$) randomly selects secret integer x_i and public identity ID_i.

x_i : a secret key of user U_i with public identity ID_i;

$y_i = g^{x_i} \bmod p$: a related public key of user U_i.

$y = \prod_{i=1}^{n} y_i \bmod p$: group public key.

Each group member uses the $(t, n \quad 1)$ secret sharing scheme to distribute his secret key to other $n \quad 1$ member.

Assuming group member U_i $(i = 1, \cdots, n)$ with secret key x_i, U_i randomly selects a $t \quad 1$ degree polynomial, $f_i(x)$, with $f_i(0) = z_i \bmod q$, and computes the secret shadow $f_i(ID_j) \bmod q$, and the related public key, $y_{ij} = g^{f_i(ID_j)} \bmod q$, for each group member U_j $(j = 1, \cdots i \quad 1, i+1, n)$.

Individual signature generation:

This new scheme allows any t group members to represent the group to sign a message m without the assistance of a mutually trusted center. Without losing generality, t group members $U_1, U_2, \cdots U_t$ represent the group and sign a message m. The t group members can sign the message simultaneously.

Group member U_i randomly selects an integer, $k_i \quad [1, q \quad 1]$, and computes $r_i = g^{k_i} \bmod p$, and make r_i publicly available through a broadcast channel. After all values are available, group member computes the value

$$R = \quad_{i=1}^{t} r_i \bmod p. \tag{15}$$

Group member U_i uses his secret key, $f(ID_i)$ and k_i, and secret shadows, $f_i(ID_j) \bmod q, (j = 1, \cdots i \quad 1, i+1, n)$, solves the follow congruence equation for integer s_i, where $m \; = H(m)$.

$$s_i R = (R + m \;) k_i + x_i + \quad_{j=t+1}^{n} f_j(ID_i)(\quad_{k=1, k \quad i}^{t} \frac{ID_k}{ID_i \quad ID_k}) \bmod q \tag{16}$$

Then, the set (s_i, r_i) is the signature for message m. Group member U_i transmits the signature to a designated clerk.

Individual signature verification:

On receiving the signature message (s_i, r_i) for message m from U_i, the clerk utilizes the public key y_i, and public key y_{ij}, authenticates the validity of the individual signature.

$$g^{s_i R} = r_i^{R+m} \, y_i (\quad_{j=t+1}^{n} y_{ji})^{\quad_{k=1,k \quad i}^{t} \frac{ID_k}{ID_i \quad ID_k}} \bmod p \tag{17}$$

If the above equation holds, the individual signature (s_i, r_i) from U_i is valid; else, the signature is invalid.

Further, the clerk uses t pairs of public values (ID_i, y_i) to construct a Lagrange polynomial function $h(y)$ as following.

$$h(y)=\sum_{i=1}^{t} ID_i \prod_{j=1, j\neq i}^{t} \frac{y-y_j}{y_i-y_j} \quad (18)$$

By the Lagrange polynomial function $h(y)$, we can use group member's public values to find the singer who signed the signature for message m.

Group-oriented (t,n) threshold signature generation:

After t individual signatures are received and verified by the clerk, the group signature of the message m can be obtained as (R,S), where

$$R=\prod_{i=1}^{t} r_i \bmod p \quad (19)$$

$$S=\sum_{i=1}^{t} s_i \bmod q \quad (20)$$

Group-oriented (t,n) threshold signature verification:

Any outsider can use the group public key y to authenticate the validity of the group signature (R,S) for message m.

$$g^{SR}=R^{R+m'}\, y \bmod p, \text{ where } m'=H(m). \quad (21)$$

If the above equation holds, the group signature (R,S) is valid; else, the group signature is invalid.

In fact:

$$s_i R=(R+m')k_i+x_i+\sum_{j=t+1}^{n} f_j(ID_i)\Big(\prod_{k=1,k\neq i}^{t}\frac{-ID_k}{ID_i-ID_k}\Big) \bmod q \quad (22)$$
$$(i=1,\cdots,t)$$

$$\sum_{i=1}^{t} s_i R=(R+m')\sum_{i=1}^{t} k_i+\sum_{i=1}^{t} x_i+\sum_{i=1}^{t}\sum_{j=t+1}^{n} f_j(ID_i)\Big(\prod_{k=1,k\neq i}^{t}\frac{-ID_k}{ID_i-ID_k}\Big) \bmod q \quad (23)$$

$$\begin{aligned} g^{SR} &= R^{(R+m')}\, g^{\sum_{i=1}^{t} x_i}\, g^{\sum_{i=1}^{t}\sum_{j=t+1}^{n} f_j(ID_i)\left(\prod_{k=1,k\neq i}^{t}\frac{-ID_k}{ID_i-ID_k}\right)} \bmod p \\ &= R^{(R+m')}\prod_{j=1}^{t} y_j \prod_{j=t+1}^{n}\Big(\prod_{i=1}^{t} y_{ji}\Big)^{\left(\prod_{k=1,k\neq i}^{t}\frac{-ID_k}{ID_i-ID_k}\right)} \bmod p \\ &= R^{(R+m')}\prod_{j=1}^{n} y_j \bmod p \\ &= R^{(R+m')}\, y \bmod p \end{aligned}$$

Signer identity verification:
If the outsider wants to determine the signer of the group signature (R, S) for message m, he can use group member U_i's public values (ID_i, y_i) to authenticate the follow equation.

$$ID_i = h(y_i)$$

If the above equation holes, the group member with public values (ID_i, y_i) is a signer of the group signature (R, S) for message m; else she is not.

5 Security and Feature Discussion

In this section, we will discuss the security and feature of new group-oriented (t, n) signature schemes. The security analysis is very similar to the one described by Agnew et al. in reference [12]. Here some new attacks are briefly examined.

(1): Recently, Tseng and Jan [7] proposed new attack on group-oriented (t, n) signature schemes, which is called T-J's attack. Any malicious attacker can compute the trapdoor information with knowing a previously valid group signature. In fact, the information is group secret key.

In the new group-oriented (t, n) signature schemes, the signature equation and the signature verification equation are the following equations.

Group signature verification equation:

$$g^{SR} = R^{R+m} y \bmod p \tag{24}$$

We can get the follow group signature equation.

Group signature equation:

$$SR = (R+m)\sum_{i=1}^{t} k_i + f(0) \bmod q \tag{25}$$

Any attacker cannot compute the group secret key $f(0)$. So, in the new group-oriented (t, n) signature schemes, any malicious attacker uses the T-J's attack method cannot generate a valid group signature with knowing a previously valid group signature.

(2): Li et al. [8, 9, 10] also proposed another new attack on group-oriented (t, n) signature schemes, which is called Li et al.'s attack. With this attack mthod, to some group digital signature and multdigital signature [6, 7, 11], any malicious attacker can forge a valid individual signature, and can generate a valid group signature with knowing a previously valid group signature.

To scheme1:

Individual signature equation:

$$s_i R = (R+m)k_i + f(ID_i)\left(\prod_{j=1, j \neq i}^{t} \frac{ID_j}{ID_i - ID_j}\right) \bmod p\,,\ (i = 1, \cdots, t) \tag{26}$$

Individual signature verification equation:

$$g^{s_i R} = r_i^{R+m} y_i^{(\,{}_{j=1,j\ \ i}^{t} \frac{ID_j}{ID_i\quad ID_j})} \bmod p\,,\ (i = 1,\cdots,t) \tag{27}$$

According to Li et al.'s attack method, let $r_i = g^U y_i^V \bmod p$, the attacker generates the follow congruence equation.

$$s_i R = U(R+m\,) \bmod q\,, \tag{28}$$

$$V(R+m\,) + {}_{j=1,j\ \ i}^{t} \frac{ID_j}{ID_i\quad ID_j} = 0 \bmod q\,,\ (i = 1,\cdots,t) \tag{29}$$

In the above congruence equation, $R = {}_{i=1}^{t}\, r_i \bmod q$, $r_i = g^U y_i^V \bmod p$, so the attacker cannot solve the equation for the individual signature s_i. The Li et al.'s attack method is invalid to the new scheme1.

To scheme2, by the similar analysis method, we also obtain that any malicious attacker uses Li et al. attack method cannot forge a valid individual signature. So, the Li et al. attack method is also invalid to the new scheme2.

(3): In 1996, Michels and Horster [13] proposed two conspiracy attacks to group-oriented (t,n) signature schemes, which is called M-H's attack. By the attack method, insider malicious attacker, say U_1, can stand in other group member's light to forge a group signature with the help of designate clerk.

To scheme 1:

Group signature verification equation:

$$g^{SR} = R^{R+m}\, y \bmod p \tag{30}$$

We can get the follow group signature equation.

Group signature equation:

$$SR = (R+m\,)\ {}_{i=1}^{t}\, k_i + f(0) \bmod q \tag{31}$$

According to M-H attack method, let $R+\tilde{m} = \tilde{R}+m$, forgery individual signature equation is following.

$$s_i\tilde{R} = (\tilde{R}+m\,)k_i + f(ID_i)(\ {}_{j=1,j\ \ i}^{t} \frac{ID_j}{ID_i\quad ID_j}) \bmod q\,,\ (i = 1,\cdots,t) \tag{32}$$

$$S\tilde{R} = (\tilde{R}+m\,)\ {}_{i=1}^{t}\, k_i + f(0) \bmod q\,, \text{ where } S = {}_{i=1}^{t}\, s_i \bmod q\,. \tag{33}$$

$$g^{S\tilde{R}} = R^{\tilde{R}+m}\, y \bmod p = R^{R+\tilde{m}}\, y \bmod p \tag{34}$$

Because of $R \neq \tilde{R}$, we have the follow inequation.

$$g^{SR} \neq R^{R+\tilde{m}} y \bmod p \qquad (35)$$

So, the group signature (S, R) for message $\tilde{m}$ generated by insider attacker with M-H attack method is invalid signature.

To scheme2, by the similar analysis method, we can also obtain the same result. The M-H attack method is invalid to the new schemes.

Michels and Horster [13] also proposed another conspiracy attack to group-oriented (t, n) signature schemes.

To the scheme2:

Group signature verification equation:

$$g^{SR} = R^{R+m} y \bmod p \qquad (36)$$

We can get the following group signature equation.

Group signature equation:

$$SR = (R+m)\sum_{i=1}^{t} k_i + \left(\sum_{i=1}^{t} x_i + \sum_{i=1}^{t}\sum_{j=t+1}^{n} f_j(ID_i)\left(\prod_{k=1,k\neq i}^{t} \frac{-ID_k}{ID_i - ID_k}\right)\right) \bmod q \qquad (37)$$

According to another M-H attack method, let $d = \tilde{m}/m$ and $\tilde{R} = R^d \bmod p$, forgery individual signature equation is following.

$$s_i\tilde{R} = (\tilde{R}+m)k_i + x_i + \sum_{j=t+1}^{n} f_j(ID_i)\left(\prod_{k=1,k\neq i}^{t} \frac{-ID_k}{ID_i - ID_k}\right) \bmod q \qquad (38)$$

$$(i = 1, \cdots, t)$$

$$S\tilde{R} = d(\tilde{R}+m)\sum_{i=1}^{t} k_i + d\left(\sum_{i=1}^{t} x_i + \sum_{i=1}^{t}\sum_{j=t+1}^{n} f_j(ID_i)\left(\prod_{k=1,k\neq i}^{t} \frac{-ID_k}{ID_i - ID_k}\right)\right) \bmod q \qquad (39)$$

,

where $S = d\sum_{i=1}^{t} s_i \bmod q$.

$$g^{S\tilde{R}} = R^{dR+\tilde{m}} y^d \bmod p \qquad (40)$$

We have the following inequation.

$$g^{SR} \neq R^{R+\tilde{m}} y \bmod p \qquad (41)$$

To the scheme1, we can also get the same result. So, the group signature (S, R) for message $\tilde{m}$ generated by insider attacker with second M-H'S attack method is invalid signature.

The two M-H attack methods are invalid to the new schemes.

(4): If an outsider intends to determine the signers, he can use Lagrange polynomial function $h(y)$ and substitutes the public key y_i to $h(y)$ and computes $h(y_i)$. If $h(y_i) = ID_i$, the group signer with public key y_i and identity ID_i is a signer of the message, otherwise, the group member is not one of the original signers. Obviously, the above verification method is right.

6 Conclusions

We have proposed two new group-oriented (t, n) threshold digital signature schemes with traceable signer based on the difficulty of solving the discrete logarithm problem. One needs the assistance of a mutually trusted center, another doesn't need the assistance of a mutually trusted center. We also discuss the security and the feature of both schemes and show that the new schemes can withstand conspiracy attacks without attaching a secret number as in the reference and can avoid to the forgery attacks proposed in reference [7, 8]. In addition, we show that, in the new schemes, any outsider can trace back to find who the signers are.

References

1. Y. Desmedt, Y. Frankel, Shared generation of authenticators and signature, in: Advances in Cryptology-CRYPTO'91, 1991, pp.457-469
2. R.L. Rivest, A. Shamir, L. Adelman, A method for obtaining digital signature and public key cryptosystem, Comm. ACM 21 (2) (1978) 120-126
3. L. Harn, Group-oriented (t, n) threshold signature and digital multisignature, IEE Proceedings of Computers and Digital Techniques 141 (5) (1994) 307-313
4. A. Shamir, How to share a secret, Comm. ACM 22 (2) (1979) 612-613
5. C.M. Li, T. Hwang, N.Y. Lee, (t, n) Threshold signature schemes based on discrete logarithm, in: Advances in Cryptology-EUROCRYPT'94, Proceedings of Eurocrypt'94, pp.194-204
6. C.T Wang, C.H. Lin, C.C. Chang, Threshold signature schemes with traceable signers in group communications, Computer Communications 21 (8) (1998) 771-776
7. Y.M. Tseng, J.K. Jan, Attacks on threshold signature schemes with traceable signers, Information Processing Letters 71 (1999) 1-4
8. Z.C. Li, Y.C. Wang, Y.X. Yang and W.L. Wu, Cryptanalysis of convertible group signature, IEE, Electronics Letter, 335(13) 1999, pp. 1071-1072.
9. Z.C. Li, L.C.K. Hui, K.P. Chow, C.F. Chong, W.W. Tsang and H.W. Chan: Security of Tseng-Jan's group signature schemes'. Information Processing Letters, 2000.Vol. 75, No. 5, Pages 187-189
10. Z.C. Li, L.C.K. Hui, K.P. Chow, C.F. Chong, W.W. Tsang and H.W. Chan: Security of Wang et al.'s group-oriented threshold signature scheme. to appear in Information Processing Letters.

11. L. Harn, Y. Xu, Design of generalized ElGamal type digital signature schemes based on discrete logarithm, Electronics Letters, 24 (31) (1994) 2025-2026
12. G.B. Agnew, R.C., Mullin, S.A. Vanston, Improved digital signature scheme based on discrete exponentiation, Electronics Letters, 1990, 26(14), pp. 1024-1025
13. M. Michels, P. Horster, On the risk of disruption in several multiparty signature scheme, in: Advances in Cryptology-ASIACRYPT'96, 1997, pp.334-345

The Implementation of Security Algorithm of Mobile Agent on Roblet*

Ying Jie Yang[1], Liang Zhu, and Fan Yuan Ma[2]

E-Commerce Research & Development Center
Shang Hai Jiao Tong University, China
[1]{yyj@ecom.sjtu.edu.cn}
[2]{fyma@mail.sjtu.edu.cn}

Abstract. The Mobile Agent Technology (MAT) is a recently developed technology with a new type of distributed computing pattern that has many advantages over ones with traditional pattern [1]. However, in order for the MAT to be widely used, the security issue must be considered. Based on the existing research of the MAT security, a feasible security algorithm for protecting an Mobile Agent (MA) against attacks from malicious hosts is proposed in this paper and the algorithm is implemented by our research group on a the MAT platform: *Roblet*.

1 Introduction

The Mobile Agent Technology (MAT) is a recently developed technology with a new type of distributed computing pattern. Compared with the technologies with the traditional Client/Server computing pattern, it has many advantages [1], for example, reducing network load, providing more scalability, and supporting network asynchronous computing. These advantages make the MAT suitable for electronic commerce applications, such as comparison-shopping, supply chain and workflow management [1,2].

In order for the MAT to be widely used, the security issue must be addressed. At present, the research efforts on the security issue of the MAT mainly concentrate on two areas: preventing hosts against attacks from a malicious Mobile Agent (MA) [3] and preventing an MA against attacks from malicious hosts. This paper mainly discusses the attack from malicious hosts to an MA.

The issue of attacking an MA from malicious hosts can be divided into two parts: computing security [4] and computing integrity [5,6]. For implementing computing security, the cost is high and many MAT applications may not need it. On the other hand, MAT applications do need computing integrity. In this paper, we focus on computing integrity by proposing a novel algorithm.

This paper is organized as follows. In Section 2, an algorithm security model state machine MA-S is proposed on the basis of the analysis of attacking an MA from malicious hosts, then the security of MA-S is discussed. In Section 3, the design and implementation of the MA-S on our MAT bed *Roblet* are presented. Finally, we conclude our paper in Section 4.

---

* This research is part of the project: *Mobile Agent in E-Business And Comparison Shopping*, supported by IBM Shared University Research (SUR) Program.

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 70-78, 2001.

2 Algorithm Design

2.1 Issue Analysis

Firstly, we simply present the implementation principle of the MAT. The MAT is usually composed of an MA and an execution environment. The MA can transfer its states and codes to another network environment to run, where the "state" refers to the attributive value that can help the MA to decide what to do when it arrives at destination hosts; the "code" refers to the program that the MA executes. Because the code of MA needs to run in different types of computer, the MAT is usually implemented with interpreting languages such as Java.

We propose an MA security algorithm in this paper, based on the security technology depicted in table 1. It implements the identification among execution environments of an MA and protects the integrity of computing of the MA code and the security of the computing result, as well as to verify whether the integrity still holds when MA is being attacked. Because the running processes of an MA in different environments are the same, the initial state of different hosts where the MA is running is the same.

Table 1. Security Technology

ENC(k,m)	Encryption is done to data m by key k
SIG(k,m)	Digital signature is done to data m by key k
MD(m)	Message Digests is done to data m

In the rest of this paper, we assume that ENC, SIG and MD are secure.

2.2 Symbol Definition

Following our analysis in Section 2.1, the MAT security issue can be summarized in the form of ordered quintuple by means of the stated-machine theory:

$$\text{MA-S} = < S\ ,\ \Sigma\ ,\ F,\ s_0,\ T\ >$$

The elements in the quintuple are defined as follows.

(1) S is a nonempty finite state set. Suppose that there are (m+1) elements in S, S={ s_i | i ϵ Z and $m \geq i \geq 0$ }. Each element in S can be described as the following vector-space:

$$S = < \text{Subject, Object ,S\_elemnt, O\_elemnt, A\_Host} >$$

◆ Subject = {host(i) | i ϵ Z and $n \geq i \geq 0$} is a set of model subjects (running environments). Suppose that there are (n+1) hosts, and 'i' is the continuous number of hosts, which presents the Object transfer direction. For example,

host(i) send the Object to host(i+1). In addition, host(0) is the host where an Object is created.

- Object = {ma} is a set of model objects (MA).
- S_elemnt = <Ks, Kp, Cert>, the security attributes of elements in Subject, is defined as below:
  - Ks ={ks(i) | i ϵ Z and $n \geq i \geq 0$}, is the private key of host(i).
  - Kp ={kp(i) | i ϵ Z and $n \geq i \geq 0$}, is the public key of host(i).
  - Cert ={cert(i) | i ϵ Z and $n \geq i \geq 0$}, is the digital certificate of host(i).
- O_elemnt = {oelemnt(i) | i ϵ Z and $p \geq i \geq 0$}, is the attributes states set of elements(MA) in the Object. Suppose that there are (p+1) states, where oelement(0) is the initial state of ma. Each element can be defined as the following vector-space:

  oelemnt = < s_cert,cert(0),st_data,sig(0),Msg,s_time,s_sig >

  - s_cert ϵ Cert, is the digital certificate of the host sending objects.
  - cert(0) ϵ Cert, is the digital certificate of host(0).
  - st_data is the static data of ma, such as codes, attributes and route information, etc.
  - sig(0)=SIG(ks(0),MD(α)), here α ╞ cert(0),st_data('╞' will be defined in subsection (2)).
  - Msg ϵ { msg(i) | i ϵ Z and $n \geq i \geq 0$ }, is a state set of data packages collected by ma, where msg(i) is the data package state processed by host(i).
  - s_time is the time-stamp created by the host that sends objects.
  - s_sig=SIG(ks(i),MD(α)), here host(i) is the sender and α ╞ s_cert,cert(0),st_data, sig(0),msg(i),s_time.
- A_host ϵ Subject is current active subject host.

(2) Σ = {i , α , β , γ , ├ , ╞} is a nonempty finite alpha lists,

- i : nonnegative integer.
- α, β and γ : variable.
- ├: variable dependence operator, eg. α ├ β,…, γ indicates that you must get all the values of β,…, γ before you get a value of α .
- ╞: variable component operator, eg. α ╞ β,…, γ indicates that α consists of β,…, γ .

(3) F = {move_ma | move_ma: $S\times\Sigma\rightarrow S$} is states transfer functions set:

```
FUNCTION  move_ma(s_i , β) {
        if(β = host(i+1))
        then
  s_cert=cert(i);
  Msg=msg(i);
  s_time=new_time; // new_time is a new time-stamp
      s_sig=SIG(ks(i),MD(α));
        α ╞s_cert,cert(0),st_data,sig(0),msg(i), s_time
      A_host=β;
 end
}
```

(4) $s_0 \in S$ is the original state and s_0 = < Subject, ma, S_elemnt, oelemnt(0), host(0)>,

- oelemnt(0) = < s_cert, cert(0), st_data, sig(0), msg(0), s_time, s_sig >.
- sig(0) = SIG(ks(0),MD(α)), here $\alpha \models$ cert(0),st_data.
- s_sig = SIG(ks(0),MD(α)), here host(0) is the sender and $\alpha \models$ s_cert,cert(0),st_data, sig(0),msg(0),s_time.
- msg(0) is secure and integral.

(5) T S is the set of security states. Every element in T must comply with the following rules:

- ◆ Condition 1: sig(0)=SIG(ks(0),MD(α)), here $\alpha \models$ cert(0), st_data.
- ◆ Condition 2: s_sig = SIG(ks(i),MD(α)), here is the sender and $\alpha \models$ s_cert, cert(0), st_data, sig(0),msg(0),s_time.
- ◆ Condition 3: Msg is secured and integral.

2.3 Security Analysis

According to the above definitions, we know that MA security is to implement the identification among hosts and protect the integrity of all parts of an MA and the security of Msg, and to identify whether the integrity still holds when MA is being attacked. To realize the object, the Certification Authority (CA) based on the Public Key Infrastructure (PKI) is required and every host needs to apply for a Cert conformed to the X.509 protocol [7]. From the security model MA-S defined in Section 2.2, we can draw several conclusions in the form of theorems as follows.

Theorem 1 *Suppose that s ϵ S, and s complies with all the three conditions in T (described in section 2.2) of MA-S model, then s is a secure state. This is to say that identity authentication among hosts can be implemented and the integrity of all parts of a ma and the secrecy of Msg can be guaranteed, and when a ma is attacked, it is easy to check whether its integrity is damaged or not.*

Proof:

◆ Host(0) Identity Authentication

- The validity of cert(0) can be proved through the CA so does host(0) identity authentication.
- Suppose host' want to personate host(0) to generate an ma.
- Because s complies with the conditions in T of MA-S model, we have sig(0) = SIG(ks(0), MD(cert(0)+ st_data));
- Because host' can not get ks(0), it can't camouflage a sig(0)' = SIG(ks(0)',MD (cert(0)+st_data)) that satisfies sig(0)'=sig(0).
- Then identity camouflage attack is prevented.

◆ Identity Authentication of hosts sending objects
It can be settled in the same way as above.

◆ Integrity of all ma's parts and Secrecy of Msg

- *Integrity of st_data*

- ✧ The integrity of st-data is basis of the overall ma's integrity, which can only be changed by host(0).
- ✧ Because s complies with the conditions in T of MA-S model, we have sig(0) = SIG(ks(0), MD(cert(0)+ st_data)).
- ✧ Because host' can not get ks(0), it can't camouflage a sig(0)' = SIG(ks(0)', MD (cert(0)+st_data)) that satisfies sig(0)'=sig(0).
- ✧ Then identity camouflage attack is prevented.

- *Confidentiality and Integrity of Msg*
  - ✧ Because s complies with the conditions in T of MA-S model, Msg is secure and integral.
- *Integrity of MA*
  - ✧ Because the integrity of ma may be attacked during the course of its moving in networks, every host is required to authenticate the integrity of ma.
  - ✧ If the integrity of ma is attacked, the attacker replaces α with α' . Here $\alpha \models$ s_cert,cert(0), st_data,sig(0),Msg,s_time.
  - ✧ Because s complies with the conditions in T of MA-S model, we have sig(0) = SIG(ks(i), MD(α'), Here host(i) is the sender.
  - ✧ Because host' can not get ks(i), it can't camouflage a sig' = SIG(ks(i)', MD(α')) that satisfies sig'=sig.
  - ✧ Therefore, the integrity authentication of ma can be implemented by host.

Theorem 2 *Suppose that s_0 is the initial state of MA-S model, then $s_0 \in T$.*
Proof: According to (4) in MA-S model and Theorem 1, we can deduce $s_0 \in T$.

Theorem 3 *Suppose that s_{i+1}= move_ma(s_i , β), where β=host(i+1), and $s_i \in T$, then $s_{i+1} \in T$.*

Proof:

Since $s_i \in T$,
we have sig(0)=SIG(ks(0),MD(α)), here α=cert(0)+st_data. (1)

From (3) of MA-S model , s_{i+1}= move_ma(s_i , β),
we have: under the condition of state s_{i+1},
s_sig=SIG(ks(i),MD(s_cert+cert(0)+st_data+sig(0)+msg(i)+s_time)).
Here host(i) is sender. (2)

Now we suppose msg(i) is secret and integral, according to (1) and (2), we can conclude $s_{i+1} \in T$.

Description: the security and integrity of Msg are analyzed in detail as follows.

In the discussion above we suppose that Msg meets the needs of security and integrity. Then how can the security and integrity of Msg be implemented? First let us analyze the construction of Msg before the further discussion.

Presume:

m(0) is a random certificated code generated by host(0).

m(1) is a data got from host(i) by ma.

M(i) is a data package that is made up of m(i) having been processed by security technology and msg(i) $\models$ M(0),M(1),…,M(i).

Therefore, to guarantee the integrity and security of Msg, in fact, is equal to guarantee those of m(i). The integrity and security of m(i) concretely means:

- ◆ Guaranteeing that the latter hosts should not get information of the formers.
- ◆ Guaranteeing that the latter hosts should not modify, remove, replace and insert new data into the msg provided by the formers.
- ◆ Guaranteeing that host(0) be able to check the integrity of each part of Msg.

In terms of the security mentioned above, the reference [8] proposes a perfect solution as below:

- *Wrapping*
  M(i)=SIG(ks(i),(ENC(ks(0),(m(i)+r(i)))+h(i))).
- *Connecting Relation*
  h(0)=MD(r(0)+host(1)).
  h(i)=MD(M(i-1)+host(i+1)).
- *Protocol*
  host(i)→host(i+1).

Referring the reference [5] to get the details and proof about this algorithm.

3 Algorithm Implementation

Roblet, designed by our research group, is a Java-based MA platform that can implement the mobile computing of Mobile Agent, on which we have designed and implemented the MA security algorithm. RSA is used in the encrypt algorithm ENC(k,m), MD-5 is used in Message Digests and the Digital Signature algorithm SIG(k,m) is implemented based on the two algorithms above. The main program flow is shown as below:

```
Boolean Roblet_MainProcess(Roblet roblet) throws InvalidRobletException
  {
   Boolean return_code;                  // returned flag

   return_code=roblet.Init();            // roblet initialization
   if(return_code.equals(FALSE)) return FALSE;
                  // roblet initialization fails and returns

   return_code=roblet.SecurityCheck();  // roblet security check
   if(return_code.equals(FALSE))  return FALSE;
                  // roblet security check fails and return
     roblet.Exec();            // roblet's code running
     roblet.Packet();          // wrap roblet
     roblet.Send();            // send roblet
     return TRUE;
   }
```

roblet.SecurityCheck is defined as below,

```
Boolean SecurityCheck()
 {
  String Cert;            // X.509 certificate
  String MA;              // MA package
  String KS;              // the public key of sending AgentHost String
  st_data;         // st_data
  String Msg;             // Msg
  String Sig;             // Sig: signature digests
  String Temp_Sig;        // temporary signature digests
  Boolean return_code;    // return flag

  // Host identity authentication
  Cert=Get_HostCert(0);   // get the certificate of a host, where 0
                 represents the original host creating roblets

  return_code=Check_HostCert(Cert)
               // check the certificate Cert's legality
  if(return_code.equals(FALSE))   return FALSE;   // Illegal, return

  Cert=Get_HostCert(1);   // get the certificate of a host, where i
                 represents the host sending roblets.
  return_code=Check_HostCert(Cert)
                // check the certificate Cert's legality
  if(return_code.equals(FALSE))   return FALSE;   // Illegal, return

  // check the whole integrity of MA
  MA=Get_MA();            // get MA package
  Sig=Get_MASig();        // get MA's signature digests
  return_code=Check_Sig(Sig);
               // check the legality of MA's digital signature
  if(return_code.equals(FALSE))   return FALSE;   // Illegal, return
  KS=Get_KS();            // get the public key of sending AgentHost
  Temp_Sig=SIG(KS,MD(MA));    // calculate MA signature digests
  return_code=Check_SigMD(Sig,Temp_Sig);
               // check the legality of MA's signature digests
  if(return_code.equals(FALSE))   return FALSE;   // Illegal, return

  // check st_data integrity
  st_data=Get_st_data();  // get st_data
  Sig=Get_st_dataSig();   // get st_data signature digests
  return_code=Check_Sig(Sig);
               // check the legality of st_data digital signature
  if(return_code.equals(FALSE))   return FALSE;   // Illegal, return
  KS=Get_KS();            // get the public key of sending AgentHost
  Temp_Sig=SIG(KS,MD(st_data)); // calculate st_data signature digests
  return_code=Check_SigMD(Sig,Temp_Sig);
               // check the legality of st_data signature digests
  if(return_code.equals(FALSE))   return FALSE;   // Illegal, return

  // check Msg integrity
  return_code=Check_Msg()                       // check Msg integrity
  if(return_code.equals(FALSE))   return FALSE; // Illegal, return

  return true;
  }
```

Where Check_HostCert(Cert),Check_Sig(Sig), and Check_SigMD(Sig,Temp_Sig) are the key functions of the algorithm. Because of the limitation of space available,

Out of these three functions, Only Check_Sig(Sig) function is described in detail here, and the other two functions have similar implementation.

```
Boolean Check_Sig(String Sig) throws InvalidRobletException
 {
  String   MA;               // MA package
  String   KS;               // the public key of sending AgentHost
  String   Temp_Sig;         // temporary signature digests

  MA=Get_MA();               // get MA package
  KS=Get_KS();               // get the public key of sending AgentHost

  Temp_Sig=SIG(KS,MD(MA));   // calculate MA signature digests
  If(Sig.equals(Temp_Sig)    return TRUE;
  Else                       return FALSE;
 }
```

In order to verify the practicability of the algorithm, we have carried on a series of experiments. The experimental environment includes three PCs, with Windows NT on one computer and windows 98 on the other two computers. The installed Java platform is JDK 1.3 software package of SUN Microsystems. In order to authenticate the identity of the AgentHost, we implemented a simulation environment of CA (Certification Authority). The experiment involves the following steps:

1. Starting up agentHost service on the two PCs (Windows 98)
2. Starting up CA service on Windows NT PC
3. Registering the two AgentHosts in CA
4. Sending Roblet from one AgentHost to another
5. Receiving AgentHost sends the request to CA for Authentication using the above algorithm
6. CA, according to the registered AgentHost, authenticates the identity. If the AgentHost is already registered, a successful signal will be returned and if not, validation will be failed.
7. Receiving AgentHost, according to the result returned by CA, carries an integrity authentication for the successful returns and for the unsuccessful returns, it returns failure signal to the sending host.

The Experiment verifies that the employed algorithm can perform the authentication of AgentHost and Integrity checking of MA. Degree of security of the algorithm is related to the security degree of encryption, message digest and digital signature algorithm. In terms of performance, there is no significant difference in transferring Roblets even after using this security mechanism during our experiment.

4 Conclusion

Based on the existing research of the security of the Mobile Agent Technology, a feasible security algorithm for protecting the Mobile Agent against the attack from malicious hosts is proposed in this paper. The proposed algorithm enhances the security of the Mobile Agent Technology.

References

1. Danny B. Lange and Mitsuru Oshima, "Seven Good Reasons for Mobile Agent", Communication of the ACM, Vol.42, No.3, pp. 88-89, March 1999.
2. Prithviraj Dasgupta, Nitya Narasimhan, Lousie E. Moser, and P.M. Melliar-Smith, "MAgNET: Mobile Agents for Networked Electronic Trading", IEEE Transactions on Knowledge, Vol 11, NO. 4 , July/August 1999.
3. Gunter Karjoth, Danny B.Lange, and Mitsuru Oshima, "A Security Model For Agents", IEEE Internet Computing, July/August 1997.
4. T.Sander, C.T. Tschudin, "Protecting Mobile Agents Against Malicious Hosts", Mobile Agents and Security, Lecture Notes in Computer Science, Springer, 1997.
5. G.Karjoth, N.Asokan, and C.Gülcü, "Protecting the Computation Results of Free-Roaming Agents", IBM Research Division Zurich Research Laboratory 1998.
6. B.S. Yee, "A Sanctuary for mobile agents", Technical Report CS97-537, UC San Diego, Department of Computer Science and Engineering, April 1997.
7. ITU Rec. X.509 (1993) | ISO/IEC 9594-8: 1995, including Draft Amendment 1: Certificate Extensions (Version 3 certificate).

Active Page Generation via Customizing XML for Data Beans in E-Commerce Applications*

Li Chen[1], Elke Rundensteiner[1], Afshan Ally[2], Rice Chen[2], and Weidong Kou[3]

[1] Dept. Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA
{lichen|rundenst}@cs.wpi.edu

[2] IBM Toronto Lab, Toronto, ON M3C 1H7, Canada
{ally|ricechen}@ca.ibm.com

[3] Dept. Computer Science and Info. Systems, E-Business Technology Institute, The University of Hong Kong, Hong Kong
wdkou@eti.hku.hk

Abstract. In this paper, we analyze enabling technologies for typical e-business applications in terms of their multi-tier system architecture and their Model-Control-View (MCV) programming model. Based on observed limitations of the JSP (Java Server Page) technique commonly adopted for dynamic page generation (the view logic), we instead propose an alternative solution approach, namely, a generic schema mapping strategy to generate XML documents and DTDs from enterprise data beans. First, we describe in detail the XML generation process for the content composition logic. We also outline the XSL processing for the transformation logic. The separation of these two logics results in a generic solution to the bean viewing problem. In particular, it improves the bean reusability via its XML representative compared to the rigid strategy of hard-coding logics into JSP. Our proposed XML mapping solution represents a potentially valuable addition to future versions of the enterprise data beans specification. The trade-off between performance and flexibility of these alternative solutions is discussed. Lastly, we survey the state-of-art research results and emerging stardards related to this XML model mapping approach.

1 Introduction

Traditional business processes are facing an unprecedented challenge in the current Internet age, experiencing pressure to turn themselves rapidly into an

* This work was conducted during the stay of the first author at the E-Commerce Development center (ECD) of the IBM Toronto Lab in the summer 2000. Li Chen would like to thank IBM for the IBM corporate fellowship. Dr. Rundensteiner would like to thank IBM for the IBM partnership award. This work was also supported in part by several grants from NSF, namely, the NSF NYI grant #IRI 97–96264, the NSF CISE Instrumentation grant #IRIS 97–29878, and the NSF grant #IIS 97–32897.

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 79–97, 2001.
Springer-Verlag Berlin Heidelberg 2001

e-business in order to stay competitive. The rapid adoption of the Internet technologies have opened a new arena for companies to compete in terms of their e-commerce solutions. Development of a comprehensive e-business solution typically involves all aspects of an Internet-based distributed system such as Java^{TM} Servlets [17], JavaServer Pages^{TM} (JSP) [18], XML [21] and Enterprise JavaBeans^{TM} (EJB) [15], as well as some programming pattern and an overall architectural structure design.

Built over a multi-tiered open architecture, the IBM WebSphere Commerce Suite [7] is one successful example of such an integrated e-business solution. It adopts several emerging technologies that together form the web application framework. The IBM WebSphere Commerce Suite exploits the Model-Control-View programming model, and it adopts JSP technique for the dynamic web page generation. However, JSP hard-codes the mapping of a bean property with its specific tags, resulting in a hence had to maintain software under either the change of the bean types or the desired XML structure. Also, this approach requires from the JSP programmers the knowledge of page design, XML schema and the specific API of access methods to data beans.

In this work, we now instead propose a generic approach for generating XML flexibly from beans. The proposed solution strategy first generates XML documents and DTDs via introspecting enterprise data beans and then completes the transformation logic using an XSL process. This two-step process for dynamic page generation is more scalable compared to the JSP solution in terms of the required development efforts. Furthermore, this approach could function as an extended XML wrapper method for any Java bean and hence relieve the programmers from the burden of implementing the composition logic every time for each new bean-XML mapping.

A real-world "purchased order" case study is conducted to compare our proposed approach with the existing state-of-the-art solution for dynamic page generation, namely, the JSP technique. One of the differences between these two approaches is whether the dynamic page generation is achieved by accessing the bean properties on the fly and then using the composition and formatting logics or via the use of introspection on beans and then the transformation or customization logics. The decision of which of these two approaches to adopt for a given application is based on the trade-off of flexibility versus performance desired for the application. This comparison also results in a broader discussion of the related technologies.

The remainder of this paper is organized as follows: Section 2 illustrates the web application enabling technologies by a high-level picture of the component-based WebSphere Commerce Suite, its programming model and some key technologies. Section 3 analyzes the requirements of XML bean wrappers. Section 4 describes in detail the proposed *toXML* function as the generic solution approach to wrapping data beans into their XML forms. The discussion about the trade-off of flexibility and performance is given in Section 5. We survey the state-of-the-art approaches of model mapping in Section 6, and finally conclude in Section 7.

2 Background on Enabling Technologies for E-Commerce Applications

2.1 Component-based Web Application Architecture

The advent of e-business, with its requirement for interoperability, has been a major driving force for the more rapid adoption of standards by the industry. As the foundation for deployment of a Web-based application, a multi-tiered architecture, as also illustrated in Figure 1, includes: (1) the tier of the Internet clients through which users can input requests and interact with Web application servers via HTTP. (2) the tier of Web application servers where the users' requests are processed to conduct any needed business logic involving access or update of the data stored in the back-end system. (3) the infrastructure services tier sitting between the application server and the backend systems where the system services are provided including the directory and security services, as well as the database connection services. (4) the enterprise critical data and computing assets residing at the tier of databases or legacy systems. They can be integrated with the application server and become the integral part of the enterprise business logic.

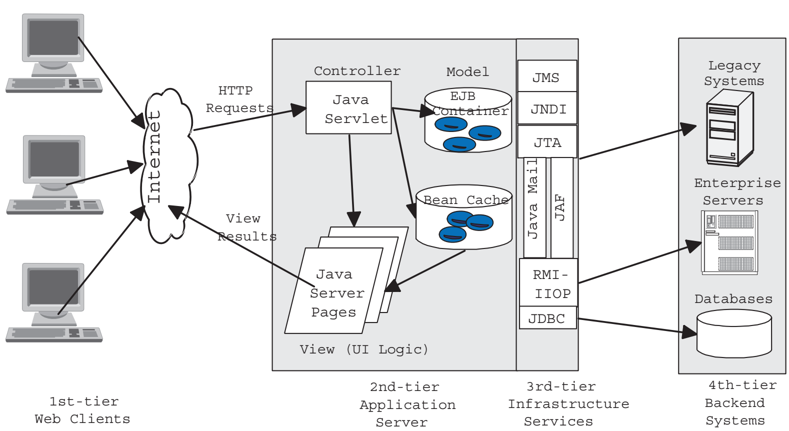

Fig. 1. Multi-tiered Web Application Architecture.

For example, the IBM WebSphere Application Server (WAS) [6] has been developed based on an open component-based software architecture to provide a comprehensive web application development environment. It adopts many standard-based technologies, de facto standards and protocols such as Java^{TM} Servlets [17], JavaServer Pages (JSP)^{TM} [18] and XML [21] techniques to be able to quickly transform static Web sites into vital sources of dynamic Web content. Enterprise JavaBean (EJB)^{TM} [15] components are used to connect and interact with the back-end relational database and to incorporate business logic. The WebSphere Application Server makes use of the auxiliary web application infrastructure services such as Java DataBase Connection^{TM} (JDBC) [9] and Java Naming Directory Interface^{TM} (JNDI) [16] to build connections to access databases and look up the services in a distributed enviroment.

Many current web sites are running using "traditional technology", such as Java^{TM} Remote Method Invocation (RMI) [19] and JDBC techniques, to manage on-line transactions without EJB technology nor the help of an application server. For different clients who may access the shared data sources (database tables) in different applications, sometimes different versions of the same code is written over and over again with the same application logics and with access to the same data sources. For this purpose, the EJB technology provides a component-based architecture for building web applications over a distributed heterogeneous object environment. It allows for the encapsulation of common data and business logic into enterprise beans, which then would take control of the access and management of the concurrent transactions and database locking. Therefore, the adoption of the enterprise bean application server would greatly improve the application implementation effort required due to the extensive reuse of data and business logics and the reduced efforts spent on the managment of concurrency, persistence and transactions.

2.2 Programming Model

Among the commercial Web application products, the IBM WebSphere Commerce Suite (WCS) [7], formerly called Net.Commerce is the industry's first integrated e-commerce hosting solution. For a variety of customers who may be involved in different business patterns (B2B, B2C, ect), WCS provides off-the-shelf functionality to help them to create dynamic e-marketplaces in a timely manner. In its development practice, the Model-Control-View programming model (refer to the 2nd Web Application Server tier in Figure 1) is adopted to structure the interaction and relationships between several major building blocks. In particular, WCS adopts Java Servlets as the interactive HTTP request controller, EJB as the bean-based accessor or connector to the back-end data model, and JSP as the server-side-include script language able to embed bean accessing code into HTML macros for dynamic page generation.

The User Interface (UI) logic of the MCV model is supported using JSP technology. JSP offers a template-based approach as typically supported by visual authoring tools. It allows for the integration of dynamic data with HTML formatting macros. In some cases, the dynamic data is obtained from the users'

input or profiles and passed to JSP as the input parameters by the Servlet controller. In other cases, the Servlet controller receives the client's request and dispatches it to JSP. If the latter needs to access a specific type of Java bean, the local cached bean representative of the remote enterprise Java bean assembles these local data beans and returns a formatted HTML page back to the client. This approach of using JSP to be responsible for the UI logic has its advantage in such a MCV model context in that its capability of accessing the encapsulated beans and of embedding them into the HTML formatting macros enables us to build dynamic web-pages.

The concept of local data beans is important since it enables fast local access to the marshalled entity bean state without the need to hit the remote EJB server whenever the application data has to be accessed.

2.3 XML/XSL Technology

Many Web application servers including IBM WebSphere are based on and support key open-industry standards such as Java, JavaBeans, CORBA (Common Object Request Broker Architecture), Enterprise Java APIs, HTTP (HyperText Transfer Protocol), HTML (HyperText Markup Language) and XML (Extensible Markup Language). XML/XSL combined technology has been justified in many application domains to be very useful for pushing data onto the web.

As XML has been recognized as the future Internet data exchange format, more and more data is stored or serialized into this format. For the Web browsers without the capability to view XML directly, Extensible Stylesheet Language (XSL) [4] provides such a facility to transform XML documents and render them in HTML. A transformation expressed in XSL describes transformation rules by associating patterns with templates. The basic process of XSL transformation is the matching the pattern against elements in the XML source and then instantiating the corresponding template to create part of the result XML.

3 Requirement Analysis of XML Bean Wrapper

Below we will now motivate why there are significant advantages for both the dynamic page generation and the XML-aware messaging systems to wrap beans into XML.

3.1 XML for Dynamic Page Generation

The JSP technique in the MCV model is primarily attractive for simple logic interactive UI scenarios where a fixed set of pre-defined presentation logics (the document content organization) and presentation view (the final HTML pages' look) can be specified beforehand. This JSP programming paradigm for dynamic page generation would not scale well in the context of providing, in a flexible fashion, a variety of customized pages. The recent emerging standards of XML

and XSL, as a newly arising solution for facilitating the development of web-based applications, now raises the question that if so how we may exploit this technology in a variety of web application servers including WCS. We therefore first propose, as an alternative solution to the JSP technique, the approach of using XML Document Type Definitions (DTD) [11] to capture the structure of the data beans' exposing properties and using XML documents as the container to be populated with bean values. Second, we propose to use XSL to transform the generated XML documents into customized presentation logic and views. However, beyond the value yielded by the use of XML as an alternative representation format to wrap Java beans, more revenue comes from the more extensive use of XML in the message exchange across different system components of WCS.

3.2 XML in Messaging Systems

To develop a comprehensive e-business application, some messaging systems need to be used to enable and simplify the integration of applications and business processes across heterogeneous back-end systems. The MQSeries [5] is being employed as the messaging system in the WebSphere Commerce Suite (WCS) to offer assured message delivery. The component of the MQSeries adaptor also acts as a broker transforming and routing messages (refer to Figure 2).

One of the main advantages of the MQSeries adaptor is that it allows the transformation and interchange between the standard MQSeries messages and the customer-defined messages, whose payload of data can be provided in XML format. Due to the availability of recent tools which can automatically reformat data to enable different applications to exchange information in the XML format, use of XML as format for encoding messages would hence greatly facilitate the system integration via such a unified "One World" interface (XML messages) with any third-party components. It also enhances the reuse of existing business computing assets from legacy systems.

For example, upon receiving such XML messages inbound or outbound from the WebSphere Commerce Server, some specific behavior within the WebSphere Commerce Suite server could be invoked, such as Customer Registration and Change Product Inventory Quantity, depending on the associated DTD .

Therefore, in a variety of application domains, XML data is becoming one of the main formats for data exchange. XML messages are being generated directly from the back-end databases or legacy file systems or are input from the front-end as the users' profiles or configurations or requests. Another significant resource of XML data exists in the form of transient messages initiated and terminated within system components.

By investigating the web application architecture in Figure 1 that is capable of hosting EJB components to implement the business logics, we found that since the enterprise Java beans, located at the 2nd tier of the web application server, as well as their representatives in the form of data beans that are local to the client code, are one of the important shareable resources. Their lifecycles exist across application transactions and they are sharable by different system

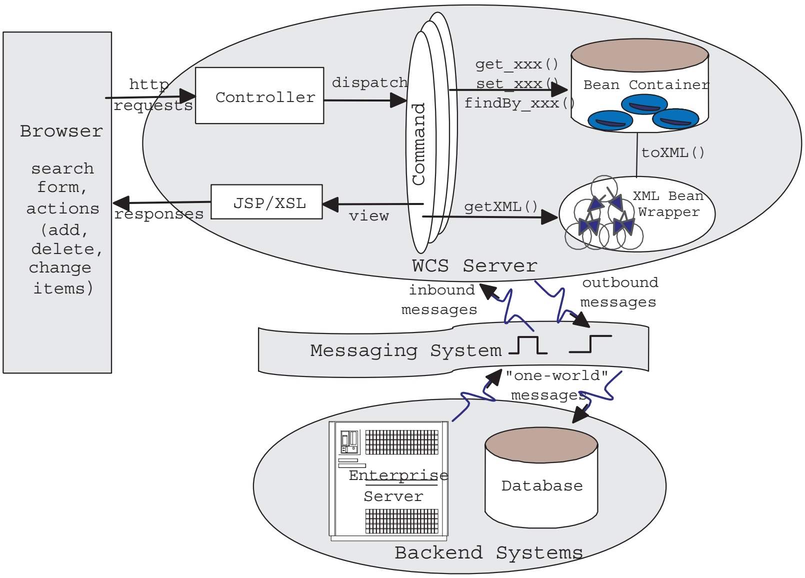

Fig. 2. Web Application System Module.

components in terms of both the persistent data being modeled and the business logics being encapsulated. Therefore, besides the XML messages provided by the back-end databases or legacy file systems or users' profiles, the data or logics encapsulated in the beans can also provide an extra resource for XML message generation. If there would be a generic approach able to automatically generate an XML wrapper for beans and then "wire" it across system components, it would facilitate the sharing of enterprise resources (data and business logics) encapsulated in beans while avoiding the performance penalty caused by accessing beans or the redundant duplication of many "local" (with respective to each single client) data beans. The generated XML messages also provide a base for further customization.

4 XML Bean Wrapper

We propose to extend the existing web application server (in general, it may be a cluster of servers) by adding an *XML Bean Wrapper* module (see Figure 2)

for the Java beans. By providing such a tool, a native XML document [1] can be generated in a generic way for a given bean, and then the XSL technique can be used to customize the native XML into the desired form.

4.1 A Scenario

In e-business, the B2C (Business to Customer) pattern makes it possible for people to do shopping on line. One common scenario is the following: Customers browse the product catalog, pick product items and put them into their shopping carts. Then they may change the quantities or even remove some of the items before finally checking them out. Upon events occurring at the client side, the corresponding requests are sent over to the web application server to invoke the business process. In an example scenario when a customer presses the "check-out" button, the process of the order starts. This process flows by performing the specific series of tasks at the application server side. For example, first UpdateInventory, then DoPayment and lastly DoShipping may occur. Here we use these names just for explanation, they are not necessarily to be precisely used in any of the real systems.

To help with the business process in such a scenario, the EJB container may host such pre-determined enterprise bean types, such as the order bean, product bean, customer bean and address bean. These beans are correspondingly mapped to some underlying data model (we use relational tables as the persistent storage, in our case). However, enterprise beans provide an object-oriented interface via which to access the shareable persistent data and perform the business logics. Therefore, different types of beans may be related to each other by the same key values and are usually bundled together to complete the tasks. For example, let's assume that the Order table is a table associated with other tables like Product_Item and Customer. Accordingly, in this OrderProcessing flow, the order bean can be considered to be a primary bean with its dependent beans reflecting an association relationship in the relational data model. Figure 3 illustrates such a mapping.

In the example scenario when an order is placed, as described above, the corresponding order bean and its dependent beans are created and their states will be made persistent in the backend data stores (either the databases or the legacy system storages) to fulfill the order placement. To provide fast access to enterprise beans by maintaining a local cache of their bean attributes and values, data beans (sometimes called access beans) are introduced. Each data bean adapts an enterprise bean (one-to-one correspondence) while hiding the remote access interface from the data bean users so that tasks can be achieved in a more efficient manner. To dynamically generate the HTML pages for the customers to view their order status, all the order related information may need to be composed, including each of the purchased product items and its shipping and billing methods. The JSP technology is commonly adopted to access the

[1] Native means that the naming convention and hierarchical structure for the tag set of the generated XML one-to-one correspond to those of the bean properties.

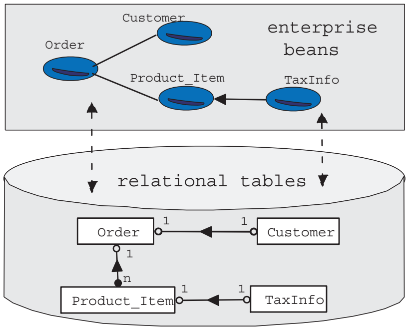

Fig. 3. Mapping of EJB and RDB.

order data beans and the other associated data beans and then assemble them by hard-coding the composition logic mixed with the presentation logic.

The round-trip process flow for this scenario is shown in Figure 4. One direction of the flow starts from the requests that are input by users. A servlet receives these requests and handles them one by one by issuing different commands and applying different business logics to the enterprise beans. Most frequently used beans are cached as data beans locally with those commands. For some requests, response pages are dynamically generated to reflect the status after the process. The latter corresponds to the other direction of the flow.

As we have introduced, the JSP technique can be used to generate a dynamic page. However, there exists an alternative path, namely, to use XML as the container for the "dumped" beans and then XSL to customize the document structure and finally render it into HTML.

The alternative solution is to bypass the JSP approach and to use the alternative XML wrapper solution, as shown in the shadow part of Figure 4. This way, the composition logic can be separated from the presentation logic and thus in many cases can be reused. The focus is then shifted to the transformation logic which fortunately can be flexibly achieved nowadays by the XSL technology.

Next, we first will describe in detail this approach and then will discuss the benefits obtained by it compared to the JSP approach.

4.2 The Bean-XML Mapping Strategy

The wrapping of any data bean into XML actually corresponds to a mapping issue between two OO-based models, namely, the specific bean class hierarchy

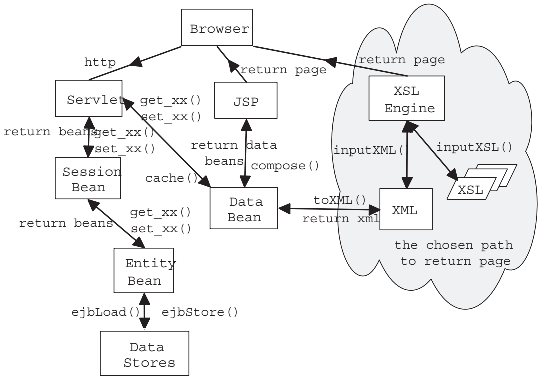

Fig. 4. The Process Flow.

and the Document Object Model (DOM) for XML, a more generalized object structure with the meta information explicitly preserved in tagnames. We now propose the following mapping strategy as described below:

1. **Map Class to DTD Type.** The type of a bean can be basically mapped into an XML DTD, namely, the root element type. However, it is not always this case especially when a dependency relationship between two bean types exists, see rule 3.
2. **Map Property to Element Type.** The bean's properties are mapped into the element types in the bean's corresponding DTD.
3. **Map Relationship to Complex Element Type.** The type of a bean property can also be of another bean type, whereby a specific association relationship is established between two bean types. Usually, at the two ends of a relationship, we would distinguish between the two beans based on which one is the primary bean in the relationship and which one is the dependent one. The dependent bean type becomes a complex element type [2] included in the primary bean's root element. Such a dependency relationship in the bean context corresponds to the foreign key relationship between the relational tables.

However, from the third mapping rule stated above, an endless loop could arise in such a mapping process if there exists any cyclic relationships between data bean types. Such a loop relationship can be detected a priori using any of the known loop detection strategy. Once the hierarchical structure (containment

[2] Complex means that the element itself has its own children elements

relationship between elements) is decided for the document by those acyclic dependency relationships, the left-over relationships (referring from the contained elements to their containing ones) can then be captured by the ID and IDREF mechanism. The ID attribute value of an XML element can always be uniquely determined by applying some naming convention to the primary key class of the entity bean (every entity bean has a key which uniquely identifies itself) from which this XML element is mapped.

To generalize the declaration for a DTD element type from its corresponding bean instances, we need to come up with an approach to determine how to structure its subelements in terms of their occurency (i.e., “once” or “*” or “+”) as well as the composition pattern (i.e., use “,” for the sequential relationship and use “|” for the choice relationship between the subelements). Since the current specification for Java beans doesn’t have the concept of optional properties, that is, a bean will always have a fixed set of properties, each of which is either of a primary type or an array of some other bean type. Therefore we can simply provide the solution using “,” as the only composition pattern. Furthermore, we always use the “once” subelement occurency for the property of the primary type while we use “+” (once or more) for a vector of properties of the complex bean type.

4.3 The toXML() Function

According to the mapping strategy we proposed above, we now introduce a function called toXML() for introspection on Java beans. This function primarily corresponds to the mapping strategy.

1. **Use getClass() to Identify Bean’s Type.** The type of a bean can be identified simply by the bean’s getClass() method, which is inherited from java.lang.Object. The obtained bean type is taken as the root element type of the result XML document.
2. **Use Bean’s Introspector and PropertyDescriptor Class to Get Bean Property Structure and Values.** An XML template and its data can be generated by introspecting on the property structure of a bean and by retrieving those properties’ values. Precisely, the name and value of a bean property are mapped into an element type and the element’s value, respectively. There is no corresponding concept in a DTD for the primary data types of bean properties such as integer. Rather they all are serialized into a string format and stored as PCDATA in XML. Such information about bean properties can be easily obtained since a bean itself is designed to expose a rich API about its well encapsulated properties in order to promote its reuse. Also, the java.beans.Introspector class and its associated java.beans.PropertyDescriptor class that come with the Java core distribution can help with the introspection on bean properties.
3. **Invoke Recursive toXML() Calls to Generate Nested XML Document.** The toXML() function starts from a primary bean and it spits out the properties of the primary types as the leaf elements, if we see the XML

document in its DOM tree structure. However, it is worth to notice that this function would invoke a recursive toXML() call on the associated bean type whenever the traversal encounters a complex property of another bean type. Thus this complex property becomes a non-leaf element. This way, chained toXML() calls triggered within the bean type dependency graph would finally result in a whole XML document with nested elements.

```
public void toXML(PrintWriter pw, Object bean) throws Exception {
  StringBuffer strBuf = new StringBuffer();
  // Analyze the bean
  Class c = bean.getClass();
  BeanInfo binfo = Introspector.getBeanInfo(c);
  PropertyDescriptor[] pdList = binfo.getPropertyDescriptors();

  // Map for each property of the bean;
  for (int i = 0; i < pdList.length; i++) {
    PropertyDescriptor pd = pdList[i];
    Class propClass = pd.getPropertyType();
    String pdName = pd.getName();

    strBuf.append("<").append(propName).append(">").;

    Method getter = c.getMethod("get"+pdName);
    // If this is a property of primary type
    if (!propClass.equals(java.lang.Class.class)) {
      Object pdValue = getter.invoke(bean);
      strBuf.append(pdValue.toString());
    }
     // if it is a vector of complex properties of bean types
    else {
      Object[] pdBeans = (Object[])(getter.invoke(bean));
      for (int j=0; j<pdBeans.length; j++)
      toXML(pw, pdBeans[j]);
    }
    strBuf.append("</").append(pdName).append(">\n");
  }
}
```

Fig. 5. The toXML() Function.

In Figure 5 we depict the main logic of the toXML() solution in its pseudo code form. We illustrate, in Figure 6, such a mapping strategy by showing the resultant XML document for an order bean instance with its dependent data beans.

We have shown that dynamic XML generation can be achieved by the toXML() mapping strategy during the traversal of the bean instance graph. A similar mapping strategy can also be applied for the DTD generation in that the latter also involves the introspection on the bean properties to determine the element type and the structual composition of its subelements. However, the DTD generation needs to generate element declarations to provide an architectural consistency for each element. This doesn't concern itself with the extraction of property values.

4.4 The XSL Transformation

Given the Java beans, their native DTDs generated using the approach proposed above do not necessarily reflect exactly the structural organization of the target

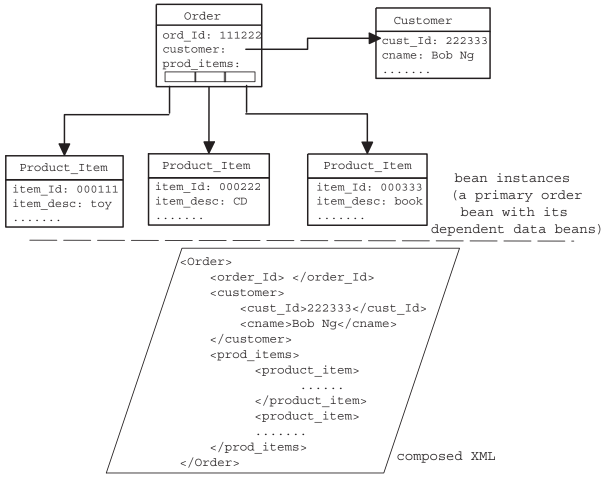

Fig. 6. An Example Mapping of Beans into XML.

documents expected by the users. Hence we now discuss how to achieve XML customizations to obtain the desired forms that satisfy the needs of a particular user. An XSL engine can help to achieve the goal.

One commonly used XSL engine is the LotusXSL [8] developed by the Lotus Co. in Cambridge, MA. It has recently been renamed to Xalan [13] and put on Apache web server [20]. An XSL processor does: 1) *Transformation*: it converts from one XML form to another, e.g., vocabulary translation, and 2) *Styling*: it prepares data for presentation, e.g., renders in HTML or any other specific presentation markup language.

In Figure 7, we give an illustrating example. The generated DTD for the orders purchased by a customer is shown on the right side of the figure. The XSL template is shown on the left side. Then the final generated purchase order form (in HTML) to be presented to the browser's user is shown at the bottom right.

4.5 Advantages

To recap, the XML bean wrapper provides the following benefits:

1. It provides a pre-packaged out-of-box solution to serialize any given data bean in a generic way, without the need of knowing beforehand the bean property information in order to hard-code the DTD mapping for it.
2. It improves the availability of the property information of data beans by making its serialization form ready to be shipped either to the client side for page rendering or within the XML-aware application components via the messaging system.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/Order">
<table>
   <tr>
      <th>OrderId</th>
      <td><xsl:value-of select = "ord_Id"/></td>
   </tr>
   <tr>
      <th>CustomerInfo</th>
      <td><xsl:apply-templates select="customer"/></td>
      </tr>
      <tr>
      <th>OrderItems</th>
      <td>
      <xsl:for-each select="Product_Item">
      <xsl:apply-templates select="Product_Item"/>
      </xsl:for-each>
          </td>
   </tr>  ....
</table>
</xsl:template>

<xsl:template match="Product_Item">
<table>
   <tr>
      <th>Item_Id</th>
      <td><xsl:value-of select = "item_Id"/></td>
   </tr>  ...
</table>
</xsl:template>

<xsl:template match="Customer">
  ...
</xsl:template>
</xsl:stylesheet>
                       Purchased_Order.xsl
```

```
<?xml encoding="UTF-8"?>
<!ELEMENT Order (ord_Id, customer, product_Items,
             status, shippingMtd)>

<!ELEMENT ord_Id (#PCDATA)>

<!ELEMENT customer (EMPTY)>
<!ATTLIST customer xlink:href CDATA #REQUIRED>

<!ELEMENT product_Items (Product_Item)*)>

<!ELEMENT Product_Item (item_Id, item_desc,
             item_features?)>

<!ELEMENT status (#PCDATA)>

<!ELEMENT shippingMtd (#PCDATA)>
                              Purchased_Order.dtd
 . . . .
```

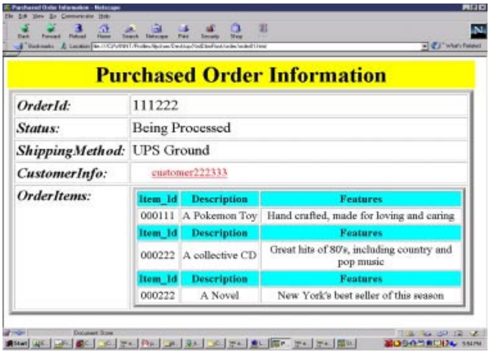

Fig. 7. The XSL Transformation

3. It separates the data composing logic from the formatting logic by using the XML wrapping function for composing XML messages out of beans and in a second phase adopting XSL for structure re-arrangement. The JSP technique, on the other hand, intermingles these two logics together and is hence a "heavy-weight" solution to dynamic page generation. Its code would be hard to maintain for non-java bean programmers.
4. It exhances the re-usability of beans via its XML representation. Due to that the format transformation tools are available both within the messaging systems and at the clients' browsers, the native XML data could be easily customized into its desired format and hence is more re-usable compared to its bean form.
5. It removes one of JSP's limitations, namely, the rendered document format is always HTML targeted for web browsers. Instead, it provides the support for the end-users to browse information on pervasive devices (such as palms or pagers). This is achieved by using XML to assemble the content while leaving the task of further transcoding or rendering to the device-specific markup languages such as WML (wireless markup language).

5 Trade-off Discussions

Since Java and EJB are still quite young technologies and their specifications are moving forward fast, the proposed XML bean wrapper approach would serve as an alternative solution to the observed problem. There are, however, additional related issues to be further investigated, two of which we will discuss below.

5.1 Considerations on Performance and Flexibility

To wrap bean properties into XML elements, the approach of toXML() using introspection at run-time pays some performance penalty to achieve this generality. That is, it is applicable to any bean to generate the corresponding XML without the need to know the bean's properties beforehand. Since the beans to be used in many enterprise applications are often relatively static, we suggest that the EJB framework is extended with the toXML() method. That is, we can add it as one of the default methods accessable via the remote interfaces of enterprise Java beans by invoking toXML() at their deployment time.

Also, a bean may provide its own customed toXML() as an overload method to the default one. For example, instead of generating an XML always capturing all of the properties of a given bean, the users may specify filters such as to expose only a subset of the bean's properties. The criteria can be set through some specified parameters in the method signature and the filtering logic can be defined in the method body.

IBM alphaworks [1] already provides the Bean Markup Language (BML) [14], which is an XML-based component configuration language customized for the Java bean component model. A bean application can be configured as described in its BML script, including the creation of new beans, accessing of existing beans, configuration of beans by setting/getting their properties, event binding, etc. The XML bean wrapper we have proposed basically functions the other way around. We hence suspect that shortcuts may exist for achieving the wrapping of a bean by making use of its BML or some kind of property profile to avoid the performance loss caused by "reflection" at run time.

5.2 The Dependency Relationship

As described in the toXML() function, we adopt the XML composition strategy to contain the XML fragment of one dependent bean into its parent bean (in terms of the dependency relationship). That is, the element type for the dependent bean type is contained as a subelement within that of the parent bean. There are certainly alternative strategies. Like the one adopted by the ReWeb system [2], where all types are treated equally and an independent XML is produced for each object, though the generated XML may be interlinked with each other via the xlinks mechanism.

There are situations when the source beans to be mapped from are isolated from each other without any relationship to enable the navigation from one bean

type to another. In such cases, different structures of them may be better captured by different DTDs and a set of XMLs rather than one single document that would currently result from the bean instances' mapping. If there exists more than one primary bean type, XML then can also be generated for the source bean set and their structures can be flatten by initiating several toXML() threads, each starting from one primary bean and navigating through its dependent beans.

A similar mapping strategy has been proposed in the ReWeb system [2] for mapping objects within object-oriented databases into XML. Again, differences of ReWeb to what we proposed here, besides the application domain, also include that the ReWeb mapping approach reflects the relationships between classes using the xlinks between generated XML documents while the toXML() solution generates one XML document to contain the serialization from all related beans.

Another issue to be addressed is that different beans may have properties with the same names. Some naming convention should be used in toXML() to avoid such a possible confusion. For example, if two beans, "BeanA" and "BeanB", both have a property called "BeanProp", some adjustment can be made to name their corresponding XML elements differently by pre-fixing their bean name instead of both using only the property name. Thereby, the XML elements generated may be "BeanA_BeanProp" and "BeanB_BeanProp" respectively.

6 Related Work

There has been a lot of research work emerging recently on the issue of modeling in XML. XML promises to be the future ubiquitous Internet data format and hence to provide a compelling environment for data to be integrated from disparate forms. The focus of the database community has been how to exploit and improve current commercial DBMS technology as the back-end storage system to manage or support XML data better on the one hand, and how to provide an automated XML schema creation facility from the traditional (e.g., relational or object-oriented) data models on the other hand. The latter issue relates closer to our current work.

Several research projects as well as commercial products focus on the problem of data transformation between XML documents and relational databases. The XML-DBMS [12] tool developed by Ron Bourret at the Technical University of Darmstadt consists of a set of Java packages providing a way to generate XML DTDs from a given database schema and vice versa. The following mapping strategy is used by their simplistic procedure to generate a XML DTD from a relational database schema: 1) For each table, create an element. 2) For each column in a table, create an attribute or a PCDATA-only child element. 3) For each primary key/foreign key relationship in which a column of the table contributes the primary key, create a child element.

In the Object-Oriented Database (OODB) context, the Re-Web project [2] at Worcester Polytechnic Institute has adopted a similar mapping strategy to construct XML elements out of objects in the OODB. A class in the OODB is

mapped to an XML element type, its attribute is mapped to the corresponding element's attribute or PCDATA-only child element and the relationship between classes is captured by the Xlink mechanism.

Such modeling work is not restricted to the database community. Rather they also arise from a wide array of application domains that seek to express their own concepts in the XML syntax. Developers are in a great need for an approach to leverage the web to exchange data between tools, applications, and repositories to enable a distributed team development environment. For example, the problem faced in the context of this work is how to create XML documents from the fine-granulated business-logic-centric software components, i.e, the enterprise java beans.

Most of the potential solutions for this bean mapping problem fall into two categories. One way is to write a certain amount of code specifically mapping individual pieces of the information from the strong typed system into the XML conforming to a pre-defined DTD. The JSP technique adopted in most commercial application servers is along this line. That is, it hard-codes the semantics mapping of the selective beans' properties and their corresponding XML tags. The other way corresponds more to a "properties dumping" of the entire bean set with the intention of re-constituting all of the beans with their relationships and properties intact. Our proposed toXML() approach bears more of this flavor. This approach suggests a more generic metadata mapping in a mechanical sense. Its strength lies in that it leverages XML concepts such as the element containment relationships and the ID/IDREF mechanism with the existing bean data structures. It hence provides one possible straightforward schema mapping.

New standards concerning the meta schema mapping keep emerging. The XML Metadata Inter-change Format (XMI) [10] proposed by OMG and IBM specifies such an open information interchange model that allow developers to exchange their programming design and data between different models or applications built in a team development environment. An XMI toolkit is also available at the IBM alphaworks site to allow the sharing of the Java application objects using XML, and the flexibility of converting designs and code between Java, UML, and Rational Rose. XMI can also be expected to be able to generate XML DTDs automatically for each type of the meta information model. The adopted approach to obtain all element and data types in XML Schema from the UML diagram is discussed in [3].

Opposite to the problem of XML schema creation from different information models, the other direction is to process XML for generating application data. This issue promises to be able to significantly improve the performance and functionality of business applications while reducing both development and maintenance costs. One of the emerging standards along this line is the Bean MarkUp Language (BML) [14].

7 Conclusions

This work is motivated by the experience we have had with the development of e-business applications. Based on the component-based architecture of the web application server, there exist some needs for an XML-centric module that is able to produce and transform XML documents customized to the needs of the end-users or the communication of the server components. We have proposed such a solution to generate XML documents from enterprise data beans. A model mapping strategy and the XSL transformation are described. The trade-off leveraged by different related approaches is also discussed.

One of the future work would be to extend the current simplistic model mapping strategy to be able to deal with more complexed situations, such as to differentiate the association relationship from the aggregation relationship. Also, a sophisticated evaluation study should be conducted to assess different solutions by a variety of criteria.

The approach to wrap enterprise data beans should be evolved with the EJB and Java specifications. As the query language for EJB, EJBQL, is appearing in the EJB 2.0 specification, we may want to re-consider the solution about how to specify the filter in a toXML() function to retrieve only the desired portion of data. Also, a good solution for inferring a DTD from bean instances needs to be set forth, and the validation of the generated DTD should be addressed.

References

1. C. Bahr, D. Jue, M. Hwu et. al. IBM AlphaWorks Site. http://www.alphaworks.ibm.com.
2. L. Chen, K. T. Claypool, and E. A. Rundensteiner. SERFing the Web: The Re-Web Approach for Web Re-Structuring. *WWW Journal - Special Issue on "Internet Data Management", Baltzer/ACM Publication, http://manta.cs.vt.edu/www*, 3(2):95–109, 2000.
3. G. Booch and M. Christerson and M. Fuchs and J. Koistinen. UML for XML Schema Mapping Specification. http://www.rational.com/uml/resources.
4. W. X. W. Group. Extensible Stylesheet Language (XSL). http://http://www.w3.org/TR/WD-xsl/.
5. IBM. MQSeries. http://www-4.ibm.com/software/ts/mqseries/messaging.
6. IBM. WebSphere Application Server. http://www.ibm.com/software/webservers.
7. IBM. WebSphere Commerce Suite. http://www.ibm.com/software/webservers.
8. IBM Alphaworks. An Experimental Implementation of the Construction Rules section of the XSL. http://www.alphaworks.ibm.com/tech/LotusXSL, 1998.
9. JavaSoft. Java Database Connectivity $(\mathrm{JDBC})^{TM}$ Specification. http://java.sun.com/products/jdbc.
10. K. J. Poole and S. A. Brodsky and G. C. Doney et. al. XMI Toolkit. http://alphaworks.ibm.com.
11. M. S. McQueen and T. Bray and J. Bosak. W3C XML Specification DTD ("XML-spec"). http://www.w3.org/XML/1998/06/xmlspec-report-v20.htm, 1998.

12. R. Bourret. XML-DBMS: Java Packages for Transferring Data between XML Documents and Relational Databases. http://www.informatik.tu-darmstadt.de/DVS1/staff/bourret/xmldbms.
13. S. Boag. Xalan: XSLT stylesheet processors, in Java and C++. http://xml.apache.org/xalan/index.html.
14. S. Weerawarana and M. J. Duftler. Bean Markup Language (BML). http://alphaworks.ibm.com.
15. Sun. Enterprise JavaBeans (EJB)^{TM} 1.1. http://java.sun.com/products/ejb/docs.html.
16. Sun. Java Naming and Directory Interface (JNDI)^{TM} Specification. http://java.sun.com/products/jndi.
17. Sun. Java Servlet: The Power Behind the Server. http://java.sun.com/products/servlet.
18. Sun. JavaServer Pages (JSP)^{TM}. http://java.sun.com/products/jsp.
19. Sun. Java^{TM} Remote Method Invocation (RMI). http://java.sun.com/products/rmi.
20. The Apache Software Foundation. The Apache HTTP Server. http://www.apache.org.
21. W3C. *Extensible Markup Language (XML)^{TM}*. http://www.w3.org/XML, 1998.

i-Cube: A Tool-Set for the Dynamic Extraction and Integration of Web Data Content

Frankie Poon and Kostas Kontogiannis

University of Waterloo, Dept. of Electrical & Computer Engineering
Waterloo, ON. N2L 3G1
Canada

Abstract. This paper presents the i-Cube environment, a tool-set that allows for Internet data and content originally available as HTML Web pages and programmatic scripts to be denoted, modeled, and represented in the form of XML documents. These XML documents conform to specific Document Type Definitions and other structural constraints that are fully customizable by the end-user or the service provider. The approach is based on representing HTML document data content in the form of annotated trees. Specific areas of interest and data content in the original HTML document that need to be encoded in the form of an XML representation, are represented as a collection of annotated sub-trees in the tree that corresponds to a large HTML document. A service integration module allows for different categories of analysis and presentation rules to be invoked according to script based user-defined logic.

1 Introduction

Over the past decade the Internet has evolved into one of the largest public communities in the world. It provides a wealth of data content and services in almost every field of science, technology, medicine, business, leisure, and education just to name a few. However, this exponential growth came at the price of increased complexity for the end-user to categorize, prioritize, and select in a customizable way the information and services that are provided by millions of Web sites across the Internet. This paper presents the i-Cube environment, a tool-set that allows for Internet data and content originally available as HTML Web pages and programmatic scripts to be denoted, modeled, and represented in the form of XML documents.

The i-Cube platform (Figure 1) is an integrated environment that provides the facilities for extracting data content from HTML in the form of a tree data structure, representing the extracted data content in XML format, and mediating the resulted XML data according to a set of lightweight service-logic. The primary goal of the i-Cube platform is to allow rapid deployment of new web-based information services (i-services) that are derived from traditional web applications, through HTML wrapping and data mediation. Secondly, the environment aims to facilitate the creation, integration, and distribution of services in a distributed Web-based environment. In this context, services are related

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 98–115, 2001.

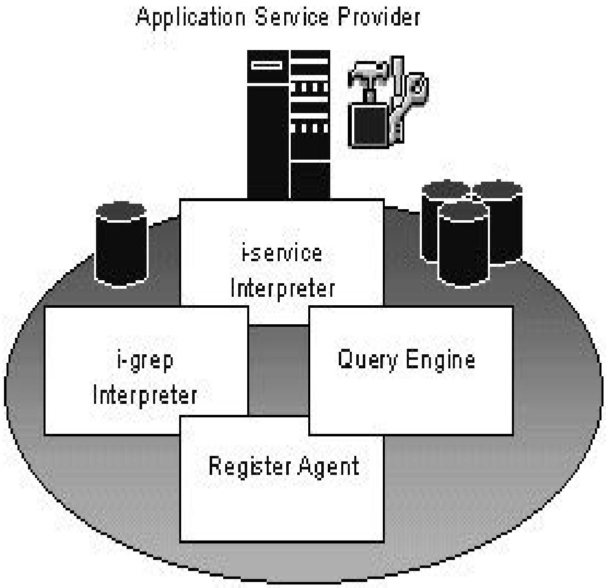

Fig. 1. The i-Cube Integrated Environment.

to extraction and processing of Web data content as this can be retrieved by specific service providing Web sites.

- **i-editor** - a graphical user interface for creating, editing, querying and registering of i-service and i-grep specifications.
- **i-grep Interpreter** (HTML wrapper) - performs extraction from an HTML source and maps the extracted data into an XML data structure, based on a set of rules defined in an i-grep specification.
- **i-service Interpreter** - (XML mediator) interprets service-logic that controls the invocations of i-grep requests and performs mediation on the data sets resulted from the invocations.
- **Query Engine** - responsible for searching i-service and i-grep specifications based on data content description, structure, and location.
- **Registration Agent** - registers i-service and i-grep specifications composed by certified remote clients in the repositories.

We define an *information grep (i-grep)* to be a user-created specification for extracting structured information from an HTML page, where the resulted data is mapped to a XML document. An *i-grep* specification consists of a location of the HTML page, a set of input parameters, an XML Document Type Definition (DTD) [18] for modeling the related data to a Web-site data content structure, and a set of data content extraction rules. An *i-grep* specification is represented in XML in order to facilitate the interchange of specifications between service providers and service builders.

Service specific logic, including the sequence of *i-grep* invocations and the manipulations of data (sorting, set operations etc) are defined in an *information service (i-service)* specification. An *i-service* specification can be viewed as a simple application that manipulates data resulted from a set of *i-grep* invocations. The specification is also represented in XML for the inherent data interchange concern. An XSLT [19] document can also be attached to an *i-service* specification for applying presentational transformation on the resulted *i-service* XML data.

We give the notions of *i-service* and *i-grep* to separate the two levels of abstraction, between service specific logics that should be processed by an application and extraction and mapping processes that should be carried out by an HTML wrapper and the mediator.

2 i-Cube Major Components

2.1 The i-Editor

The i-Cube editor is a front-end graphical interface that allows users, ranging from web developers to end-users of the web, to create extraction rules, define XML representation of Web data content and associated mapping and processing rules, using a declarative approach. The editor is written in Java, and is essentially an HTML browser with added features for creating *i-service* and *i-grep* specifications. The design goal of this editor is to provide users the exact look-and-feel of their web browser, such that they can easily select information and data content that is of their interest and should be extracted from the HTML page they currently browse.

The editor is based on a parser that parses an HTML document (fetched from an URL) and renders it for displaying on the editor's browser. At that point, a user can create an i-grep specification by highlighting HTML segment of their interest. The screenshot in Figure 2 illustrates the rendering of the HTML source as this is received by the Web-service (in this case the Canadian bookstore Chapters.ca). The significant difference between the editor and a Web browser is that the editor allows the user to highlight specific portions of the rendered page and represent this highlighted segment as a tree based data structure. The editor will assist on modeling specific data content in a fully customizable way for the user. Note that once a data entity is modeled, similar HTML pages (i.e. information for another book) that originate from the same Web service provider can be analyzed automatically.

The i-grep editor takes two steps in its initialization. First, the URL is parsed to obtain information on the protocol, host, path, application and input parameters required. Second, a *basepath* (see Section 3) is formulated to uniquely locate the selected data content in the original HTML document. This path forms the basis of our HTML *hierarchy-based* extraction approach. Next comes the declarative process for defining how data elements are to be extracted and mapped. The user can select a text node from the HTML tree and specific a name for the data element (e.g. the text $63.70 that is denoted as a Book/Price data element), which leads to the generation of a <`Rule`> tuple consisting of an extraction rule

Fig. 2. Screenshot of the editor browser.

and a mapping rule (see Figure 3). Based on the *basepath*, an extraction rule with the path `tr/td/table/tr/td/font` is generated and is associated with a mapping rule to the data element `Book/Price`. Manual editing of the generated rules user is also allowed.

Upon completion of the editing, an XML DTD is inferred from all the specified data elements. The resulted i-grep specification will be saved to a repository locally or through the registration agent to a remote repository. Such an i-grep specification is illustrated in Figure 4. Note that the inferred DTD will also be stored at the corresponding site, and the i-grep specification will include a reference to this DTD through a URI.

Composed i-grep specifications are added to in the i-grep specification tree in the Editor's main window. The user can then add individual i-grep specifications to an i-service specification, and choose the type of service-logic associated with the i-grep invocations. Data specific logic such as sorting and set operations can be applied on the i-grep result sets. The current prototype requires the user to insert an operation tag (e.g. `<Sort>`) to encapsulate the set of participating i-grep invocations. The modeled i-service specification can be saved to a local or to a remote repository through the registration agent. Finally, the editor also provides the necessary user interfaces for querying and registering data content specifications from various sources (see Query Engine and Registration Agent Sections for a more detailed discussion).

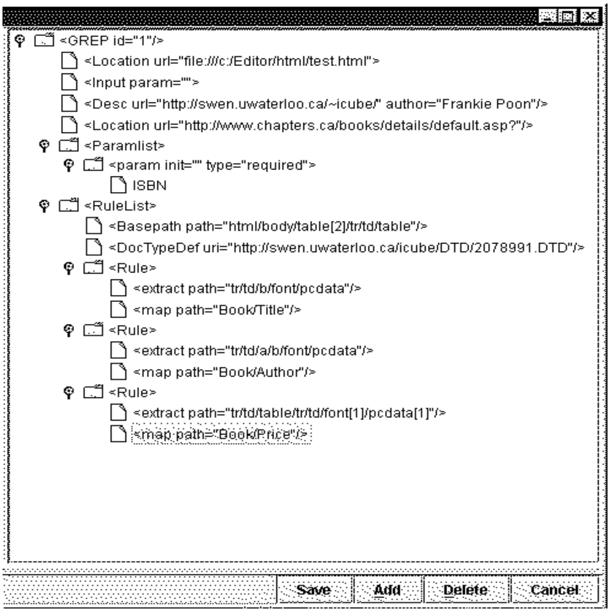

Fig. 3. Screenshot of the i-grep editor window.

2.2 The i-Grep Interpreter

The *i-grep interpreter* is essentially an HTML wrapper that encapsulates the functionality for modeling, extracting, and mapping HTML data content according to an XML-based i-grep specification. The interpreter consists of an Extraction and Mapping Rule Engine (Figure 5). As inherited from an HTML wrapper, its primary responsibility is for retrieving an original HTML document and applying extraction rules on the document. The Mapping Rule Engine then applies the mapping rules for constructing the final XML data structure from the data extracted.

The *i-grep interpreter* uses an XML parser to parse the i-grep specification, and realizes the locations and input requirement for fetching the HTML page. The HTML page is then processed to output the final XML data structure.

The result set of every i-grep invocation is stored in a session data object, belonging to a particular i-service request, managed by the Session Manager. The *i-service interpreter* can apply operations to manipulate data among the i-grep result sets stored in the session object. The extraction and mapping process is discussed later in this paper.

2.3 The i-Service Interpreter

i-service Interpreter. Control Mediation. The *i-service Interpreter* (Figure 5) is responsible for applying higher-level logic for controlling i-grep invocations, and second, it performs data manipulation on result sets resulted from

```
<?xml version="1.0"?>
<GREP id="2078991">
<Desc url="http://swen.uwaterloo.ca/~icube/" author="Frankie Poon">
    Chapters ISBN Search Engine
</Desc>
<Location url="http://www.chapters.ca/books/details/default.asp"
type="app"/>
<Paramlist>
    <param init="" type="required">ISBN</param>
</Paramlist>
<RuleList>
    <Basepath path="html/body/table[2]/tr/td/table"/>
    <DocTypeDef uri="http://swen.uwaterloo.ca/icube/DTD/2078991.DTD"/>
    <Rule>
        <extract path="tr/td/b/font/pcdata"/>
        <map     path="Book/Title"/>
    </Rule>
    <Rule>
        <extract path="tr/td/a/b/font/pcdata"/>
        <map     path="Book/Author"/>
    </Rule>
    <Rule>
        <extract path="tr/td/table/tr/td/font[1]/pcdata[1]"/>
        <map path="Book/Price"/>
    </Rule>
    <Rule>
        <extract path="tr/td/a/b/font/pcdata"/>
        <map     path="Book/Delivery"/>
    </Rule>
    <Rule>
        <data type="cdata">http://www.chapters.ca</data>
        <map  path="Book/URL"/>
    </Rule>
</RuleList>
</GREP>

2078991.DTD
<!DOCTYPE Book [
<!ELEMENT Book(Name,Author,Price,Delivery)>
<!ELEMENT Title(#PCDATA)>
<!ELEMENT Author(#PCDATA)>
<!ELEMENT Price(#PCDATA)>
<!ELEMENT Delivery(#PCDATA)>
<!ELEMENT URL(#CDATA)>
>
```

Fig. 4. Example of an i-grep specification and the inferred DTD.

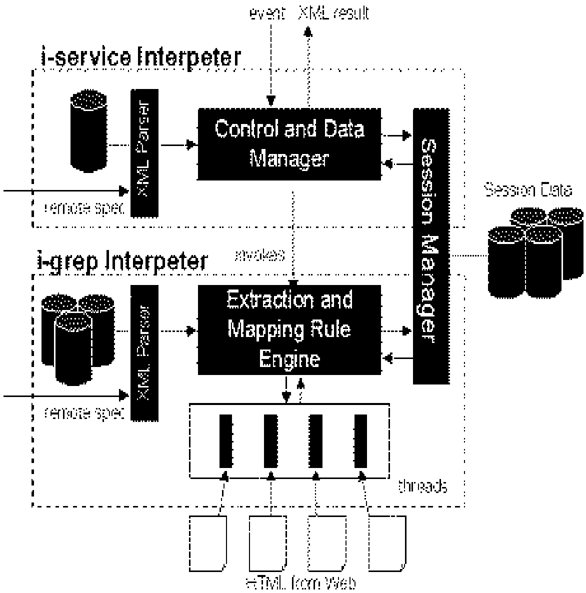

Fig. 5. The i-service and i-grep interpreter.

the invocations. In this way, distributed services located virtually everywhere in the world, can be combined on as required basis using script-based transaction scenarios, forming thus collaborative systems [6], [7]. The customization of the transaction and integration logic required by various processes to complete complex services, opens new opportunities in Web-enabled e-Commerce and e-Business environments. In this sense, business partners can customize their business transaction process models to fit specific needs or, specific contract requirements. This customization is transparent to third parties, and provides means to complete business transactions accurately and on-time. Organizations can enter the e-Business arena by building and deploying extensible and customizable services over the Internet using existing data content that can be delivered over the Web and is readily available as a service over the Internet. Moreover, virtual agencies that provide a wide range of services can be formed by integrating existing functionality and content over the Web. For example, a virtual travel agency can be formed, by composing in a customized manner, services that are readily available in various travel related Internet Web sites. Client processes may post requests to the virtual agency. The agency can enact its transaction logic (scripts) in order to integrate and compose data and services from a wide spectrum of sites. In this scenario, data about pricing, availability and, travel related special offers, can be fetched by various sites, processed by the agency and presented to the client in a customized and competitive way for the agency.

Data Mediation. Every invoked i-grep (HTML wrapping) stores its resulted data in a session object, corresponding to the current i-service request. This

allows the i-service interpreter to apply data integration related operations on the result sets, such as sorting and set operations. Our current prototype is able to sort and apply or operation to some i-grep result sets. Figure 6 illustrates an example of how a sort operation using in an i-service specification can be defined on the resulting data content (Book/Price) extracted from various sites.

```
<?xml version="1.0"?>
<SERVICE id="3453342">
<Desc url="http://swen.uwaterloo.ca/icube/" author="Frankie Poon">
          Meta ISBN Book Search Service
</Desc>
<Location url="http://www.chapters.ca/books/details/default.asp"
          type="app"/>
<Paramlist>
    <param type="required" initval="">ISBN</param>
</Paramlist>
<Sort order="ascend" key="Book/Price">
<DocTypeDef uri="http://swen.uwaterloo.ca/icube/DTD/3453342.DTD"/>
    <grep id="2078991"/>
    <grep id="4581155"/>
    <grep id="901546"/>
    <grep id="24159757"/>
</Sort>
</SERVICE>
```

Fig. 6. Example of an i-service specification.

In addition to manipulating i-grep result sets, the i-service interpreter has a special feature for *promoting* a particular i-grep service. Section 2.6 discusses in more detail the different roles of i-Cube clients. In short, this feature gives end-users the flexibility of invoking more formally defined i-services and i-grep specifications, created by either the service provider or the original web site. This feature involves sending a query request to registered i-Cube platforms, specifically to their *Query Engine*, for finding services that match a particular data structure, location and description. Once a specification is matched, the i-service interpreter either pulls the new specification from the remote site, or requests the invocation of the specification at the remote site.

2.4 Query Engine

The *Query Engine* of the i-Cube platform provides a searching facility for specifications that match the description, data structure and URL specified in a query request. This module is beneficial to end-users who wish to reuse existing specifications. Moreover, end-users can look for a more formal specification, either at the service provider site or the original web site, to replace the one that they have defined previously.

Specifications that match the query constraints are either returned to the user as an XML document, or invoked at the i-Cube platform where the specification resides. In general, the Query Engine offers two classes of query services. *Explicit Querying* is used by end-users who wish to compose their own i-service specifications and who want to know if formal specifications have been previously defined by the service providers or the original web site. This type of querying is performed by end-users through the Editor user interface.

The second type of querying service is called *Transparent Querying*, which is primarily used by the i-service interpreter. Composers of i-service specifications can indicate that they would like to invoke more formal i-grep specifications defined at the service provider (or original web site), if such specifications exist. A query request is sent to the corresponding remote query engine, if a specification exists at the remote site, then the interpreter will discard the local specification and invoke the remote one. This scheme is also known as *promoting* as mentioned in previous section.

2.5 Registration Agent

The *Registration Agent* provides an interface for remote certified i-Cube clients to register composed i-service and i-grep specifications to the repositories. This allows clients who do not possess the full i-Cube platform to submit customized i-Cube specifications to a service provider. The service provider in turn, will provide the application platform for processing their specifications upon request of the owner of the specifications.

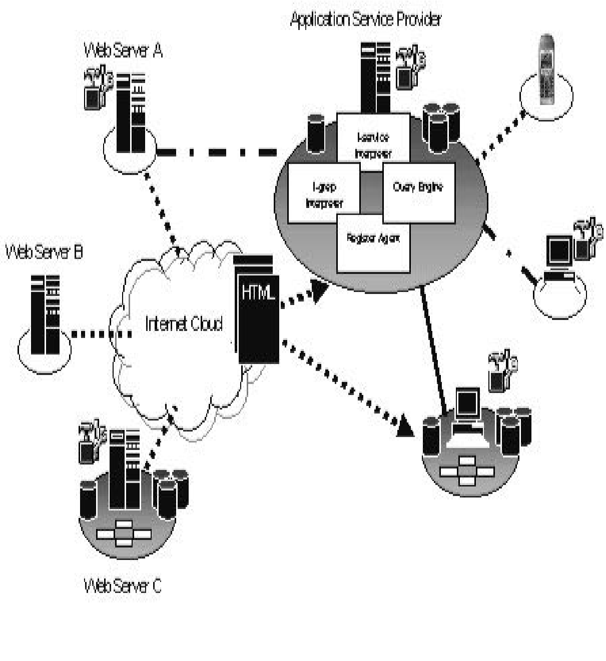

Fig. 7. The i-Cube clients and their operating environment.

2.6 Roles of i-Cube Clients

We now focus on demonstrating the different roles that are played by different i-Cube clients and their corresponding operational environments. In Figure 7, three types of i-Cube clients, service provider, end-users and original web sites are illustrated.

Service Provider. A *service provider* is a dedicated i-Cube platform that offers i-Cube based services to end-users. In addition to enhanced web services that it developed for end-users, it also allows i-Cube certified clients to register their composed specifications through the registration agent, providing them an application platform for invoking these services. In the figure, the application *service provider* that offers enhanced web services to end-users represents the service provider. The service provider consists of all five of the i-Cube components described in the previous section.

Internet clients such as handheld devices and desktop users can use the wide selections of enhanced web services offered at the site, which are represented as a typical HTML page. An example will be an HTML form that contains a Textbox and a Submit button for an enhanced search engine of books from various online bookstores where the form corresponds to a particular i-service specification. Once the user has submitted the HTTP request, the service provider carries out the appropriate service-logic, invokes the corresponding i-grep specifications, processes the result sets, and presents the resulted data in HTML format. Following the previous book example site, it means that the list of book titles is returned to the user as an HTML page.

Web Site. A *web site* is where the HTML information originates. For non i-Cube enabled web sites, it is transparent whether enhanced web services have been derived from their web applications. It is certain that if enhanced services are to put into a commercial use, legal agreement must be established between the service provider and the web site. On the other hand, for web sites that provide their own enhanced web services to the users, but do not wish to apply major changes to their existing infrastructure, they can adopt the i-Cube platform as the wrapping agent for rapid deployment of new web-services. Web sites can also compose their own specifications and register them at the service provider site, making the services available to a wide group of end-users.

End User. *End users* correspond to the users of these enhanced web-services. In the figure, the handheld device and the desktops are classified as end-users. Typical clients, like desktop users (non i-Cube enabled) can use these services by navigating on the service provider web site, where different selections of web-services are presented to them either as hyperlinks or input forms, in HTML format. The type of services will be transparent to the clients, in terms of whether they are i-Cube based or not.

On the other hand, i-Cube-enabled clients are permitted to compose their own specifications or build on existing specifications offered by service providers and web sites. They can either register their composed specifications at an i-Cube-enabled service provider, or invoke these services locally.

3 Extracting XML Data Structure from HTML

3.1 Overview of the i-Grep Interpreter

The i-grep Interpreter uses an XML parser to read information contained in an i-grep specification. It forms a complete URL request by attaching the URL and the input parameters, if appropriate, for fetching an HTML document. To assure that the document is well formed, it is being cleaned up using the combination of a HTML parser and HTML Editor Toolkit from the Java Swing class. Then according to the extraction rules specified, the extraction rule engine extracts information, and maps each piece of information to the corresponding element in the XML data structure.

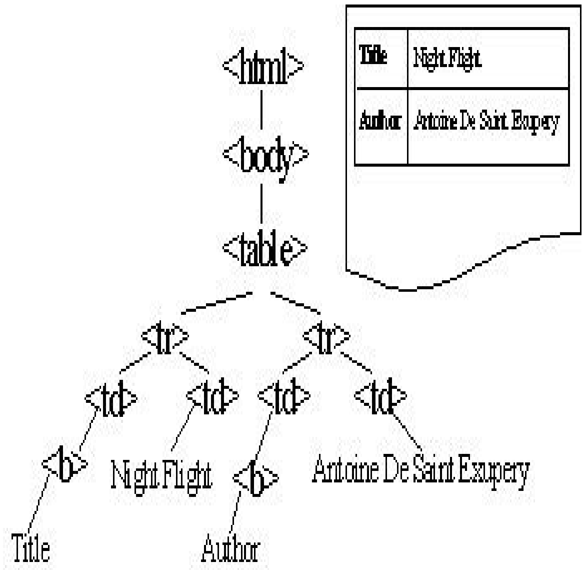

Fig. 8. An example of HTML hierarchy.

3.2 HTML Hierarchy

An HTML document is a structured document that contains text and hyperlink data stored under different presentational tags. An HTML document can be parsed by an HTML parser to construct a DOM tree that corresponds to its HTML hierarchy, such as the one in Figure 8. Each node in the figure corresponds to an HTML tag, where each leaf node corresponds to a PCDATA node for text chunk, or an empty tag such as <BR> or <IMG>. The significance of this tree hierarchy is that an HTML *tree path* can be derived to locate a particular node in the hierarchy.

3.3 Extraction Approach

Our extraction approach takes advantage of the tree hierarchical structure that is used for denoting HTML data content in a Web page in order to localize and extract specific data elements from the HTML document representing such a page. We first define the concept of an *absolute HTML tree path* to be a path for locating a specific node on a HTML tree. This path adopts similar features in XPath [20] to locate elements on a HTML tree, starting from the root node to a specific node on the tree. Using a Document Object Modeling (DOM) implementation, it is straight forward to navigate on the HTML tree along the path, `html/body/table/tr[0]/td[1]/pcdata` in order to obtain the text data *"Night Flight"* from the HTML document in Figure 8.

Each HTML element that should be traversed is separated by the character "/" to represent a parent-to-child relationship in the tree hierarchy. Identical children are indexed with respect to their sequential appearances at the parent node. The syntax `element[id]` is used for this purpose to indicate a child node at a specific index, such as `td[0]`, `td[1]`, `td[2]`, etc.

Once the corresponding HTML node is located, three types of data can be chosen for extraction: PCDATA (Parsed Character DATA), CDATA (Non-parsed Character DATA) and attribute values. They are specified at the last element in the path as `pcdata`, `cdata`, and `element(@attributename)`

It should be noted that the HTML hierarchical path alone is only effective at extracting static HTML sources that are not subject to change, and sources with no repeating data structures (e.g. 7 matches resulted from a book title query). To extend our extraction approach - such that it can provide some degree of flexibility when extracting data from a modified source, and can identify and extract one or more data structures contained within the HTML document – we include the notion of *keyword definition*. Keywords are string constants associated with a data structure that carry some contextual meanings and are defined during the creation time of i-grep specification. For every keyword defined, its corresponding HTML hierarchical path, with reference to the basepath, is computed and a conditional expression is formulated based on the keyword constant `element/pcdata=keyword`

3.4 Extraction Process

Figure 9 illustrates the rule listing of an i-grep specification that corresponds to extracting the data content from Figure 8. The extractor first locates the *basenode* that contains the descendants constituting the data structure, which is specified by the *basepath*.

After locating the *basenode*, the extractor applies a validation process to examine whether the structure located under this node is valid according to the two criteria: HTML hierarchy and keyword definitions. Each data to be extracted is defined as an individual <`Rule`> tuple consisting of the HTML hierarchy and one or more keyword definitions. If the extractor has successfully located the keywords and the data according to the HTML hierarchy, then it proceeds to the next <`Rule`> tuple. If however, the extractor has failed to locate the *basenode* in

```
<RuleList>
    <Basepath path="html/body/table[0]"/>
    <Rule>
            <extract from="tr[1]/td[1]/pcdata"/>
        <keyword from="tr/td[0]=Title"/>
        <map to="Book/Title"/>
    </Rule>
    <Rule>
        <extract from="tr[2]/td[1]/pcdata"/>
        <keyword from="tr[2]/td[0]=Author"/>
        <map to="Book/Author"/>

    </Rule>
</RuleList>
```

Fig. 9. The rule listing of an i-grep specification.

the initial stage, or failed to locate the HTML hierarchy and keywords associated with the data structure, then it will declare the process as a failure.

A recovery phase will proceed in the event of a failure. The recovery phase assumes the existence of the data structure – that is, it preserves the rules contained in a <Rule> tuple – but under a different *basepath*. The extractor then begins a search on the HTML tree, starting among the sibling nodes of the *basenode*, in an attempt to locate the same data structure specified. The extractor continues to search at the ancestors and descendants one level at a time, starting at the *basenode level*, until a matching structure can be found.

A successful recovery phase updates the specification by inserting a new basepath to the rule list. This provides a recommended alternative for the extractor to locate the data structure before a recovery phase should take place. The new basepath will become the primary path while the original one will become the secondary of the searching criteria.

In many cases in addition to using the exact HTML hierarchy to locate specific data content that is to be extracted, it is also necessary to match specific larger portions of the HTML structure. This requires techniques in applying approximate or partial pattern matching that can be based on regular expressions or stochastic techniques. Work on regular expression based pattern matching has been investigated within the context of bio-informatics, image processing, and in the area of compiler optimization [1], [5], [14].

3.5 Mapping to an XML DTD

Once the data structure is successfully extracted, it is assigned to an interchangeable format. For this purpose, we adopt Extensible Markup Language (XML) as the interchangeable format for the extracted data. Mapping rules are created at i-grep specification creation time. Similar to our extracting rules, we use a notation analogous to XPath to designate the mapping relationship between

the extracted data set and the corresponding XML Document Type Definition (DTD) that models the domain at the specific Web site and service (i.e. bookstore, travel agency).

A mapping rule corresponds to a many-to-one relationship between extracted data and an element in the XML DTD. It can contain one or more extraction rules, where all the extracted data (string based) are concatenated together, with each separated by a white space. Following the example, a rule for mapping the extracted book name to the `<Name>` element in the DTD is illustrated in Figure 9.

4 Applications

4.1 Building a Meta Book Search Engine

We now illustrate a specific example of using our integrated environment for building a meta book search engine – a service for comparing the prices of a book from different online bookstores. Using the editor, any user can create an i-service based on a set of i-grep specifications for checking the price of a book from various online bookstores. Most online bookstores offer a service for searching a book title based on its ISBN. To simplify our example, let's assume our meta search engine offers the same service, but for five different web sites. From the editor's browser, we visit an online bookstore, for example, Amazon.com and use its advanced search engine to look for a specific book title. The page showing in Figure 3 is actually the result of submitting the ISBN query "1861004044" on its advanced search engine facility, which displays the book title "Professional WAP" and its related information in HTML.

In this context, we are interested in creating a service that displays the *name* of the bookstore, the *title*, *price*, *availability* of the book as well as an *URL* link to the result page, in order to achieve so, we highlight on the browser the portion that corresponds to these information and create an I-grep specification based on the selection. The editor then brings us to a declarative process for defining the data structure that corresponds to the selected portion as shown in Figure 4. As an example, we define the element *title* and associate this element with the text `Professional WAP` contained in that page. The editor, in turn, will generate the corresponding extraction and mapping rules for this data element. We can fill in the name element with the appropriate bookstore name. A small description about this specification can also be included, like `search book title using ISBN`.

Since all ISBN searches for books on a specific site will generate HTML documents with identical structure, this mapping will work for all HTML pages generated as a result of an ISBN book search. It becomes then apparent how this process can be applied similarly to other online bookstores as in Figure 9. The result is a set of i-grep specifications that takes a common input parameter (ISBN) and its output conforms to our book information domain model format Book:bookstore,title,author,price,delivery,url. This is feasible because the mapping conforms to a domain model (i.e. DTD) for a given type of service (i.e. book store) results obtained from various other sites can also be mapped to the same

domain model and DTD. This allows for direct algorithmic processing such as sorting the obtained results by a specific field. In this context, domain models and DTDs for other services such as airline ticketing or B2B transactions can be easily built and customized to accommodate specific post-processing requirements.

The next step in the process is to compose an i-service specification based on these i-grep specifications, such that we can apply control logic to the i-grep invocations and also apply data manipulating operations such as sorting in this case, on the resulted i-grep data sets. All the specifications created are then registered at the service provider's repositories.

To access the new service, assuming we have created a very simple HTML entry page that contains an HTML form that consists of a Textbox and a Submit button. We can type in the desired ISBN number and perform a meta search on all the associated online bookstores. The backend processing will involve fetching the corresponding i-service specification and associated i-grep specifications and invoking them respectively. Each i-grep invocation will extract the book data from a HTML web site, based on an ISBN, which is then stored in a Session Data Object at the remote site. The result sets are then sorted according to their prices as defined in the i-service specification; the final data structure is an XML document that encapsulates all the book data called `Booklist:`{`Book`}

A default XSLT template is provided by the service provider for transforming the resulted XML data set into HTML, then we will be able to see the list of books, including the bookstore, title, author, price, availability and an URL link to the bookstore presented in a HTML table format as shown in Figure 10.

4.2 Notification Service

A service can be created for monitoring changes on a web site [13], and the resulted information can be sent to the end-user through email, or other types of notification mechanism such as the PUSH mechanism in WAP [17]. To illustrate a practical example, assume an online library book search system that returns to the user the information of a book, including its name, author and status. A book that is currently on loan will display that status information: `Status: On Loan`, user can make use of this information and define a service for monitoring changes of the status until the book becomes available, for example `Status: In Library`. At that time, the service will compose a notification email indicating the availability of the book. All it requires is that the service periodically analyze the HTML page for the user.

4.3 Migrating from HTML to VoiceXML and WML Applications

The introduction of new presentational markup languages not only provides tailored presentational means for different types of devices, but also at the same time aims to attract a new domain of web users. VoiceXML [16], for example, aims to attract users who are familiar with interactive voice system; similarly, WML is designed for mobile handsets that adopt the WAP standard [17]. To migrate existing HTML pages into these markup languages, it is possible to

Your Results: ISBN 1861004044

Store	Name	Author	Price	Availability
Fatbrain.com	Professional WAP	Charles Arehart, [more]...	$41.95	In Stock - Orders received by 4p.m. ship the same day
bamm.com	Professional WAP	Charles Arehart, [more]...	$43.19	In Stock: Ships with 2-3 days [more]...
BN.com	Professional WAP	Charles Arehart, [more]...	$47.99	In Stock: 24 hours (Same Day) [more]...
Amazon.com	Professional WAP	Charles Arehart, [more]...	$47.99	Usually ships within 24 hours.
Borders.com	Professional WAP	Wrox Press Inc., [more]...	$50.00	This title usually ships in 24 hours.

Fig. 10. Book price comparison results.

use transformational language such as XSLT (eXtensible Stylesheet Language for Transformation) to define one-to-one, or one-to many mapping between the elements in two markup languages. However, it becomes a very difficult task for the XSL developer to write rules for the rather long and complex structure of HTML. It will be ideal if significant segment of the HTML page is represented in a XML format that carries contextual meaning (data structure) using the i-Cube editor, such that the XSL development process will become more obvious to the developer.

5 Related Work

Different approaches for extracting information from HTML sources have been recently developed. While many approaches focus on the definition of a formal and expressive extraction grammar [8], [11], and [12], others [2], [3] focus on an SQL-like language for querying semi-structured information from HTML sources. The inherent problematic nature of writing extraction rules manually has motivated the development of graphical tool to assist the generation of extraction rules. In This context, W4F [15] uses a similar hierarchical-based navigation approach for extracting structure from HTML. Its extraction wizard provides a graphical interface for visualization of annotated HTML segment to assist the writing of extraction rules.

Other approaches take a step further to enhance the extraction rule generation process by incorporating an inductive learning mechanism [9] and using a two-phase code generation framework for the wrapper generation process. A micro-feedback approach is also used in order to customize the generated wrapper at runtime.

It should be noted that many of the explored approaches taken, like those presented in [4], [9], and [12], are aimed at the generation of a set of extraction rules that is fed to a code generator for building a standalone wrapper application. Our approach is different in that we focus on the building of an interpreter that understands extraction and mapping specification, in addition to the location of the HTML pages. We adopt the XML technology in our interpreter design to encourage the exchange of wrapper specifications over the Internet, which in turn, allows derived information services to expose to a wider domain of web users. Finally, work that deals with control integration aspects of various distributed services using scripting languages, such as the Event-Condition-Action (ECA) framework has been presented in [10].

6 Conclusion

In this paper we have presented the i-Cube integrated environment and demonstrated its capabilities for rapid deployment of web-based information services to end-users by wrapping existing HTML pages and reusing data structure that they provide. We have also presented the extraction and mapping rules that are used for precisely extracting data from HTML into a corresponding XML data structure. In particular, the extraction rules that we propose make use of the inherent tree characteristic of HTML hierarchy, which allows for the denotation and localization of data content based on an HTML tree path. At the service level, a prototype system that allows for the definition of the control logic of invoking i-grep specifications and operations that pertain to specific data mediation processes.

The integrated environment provides a set of tools for creating, customizing, invoking, registering and querying of composed specifications. This makes i-Cube an environment, not only suitable for the commercial Web sites, but also suitable for service providers who wish to provide to end-users a new domain of web services derived from existing ones. Moreover, end-users can benefit from using the tool for creating customized web services, which works in conjunction with a registering mechanism for storing their composed specifications at a remote service provider, serving as an application platform for the end-users. From an end-user perspective, the environment provides a common platform for deploying customizable web services tailored to user's own needs. Existing web systems, on the other hand, can easily realize the added benefit of deriving and deploying new domain of web services without the expense of applying changes in their existing infrastructure. Finally, the service provider who acts as the broker of the entire architecture plays the important role of driving the process of distributing web services among web systems and end-users.

References

1. A. Aho, M. Ganapathi, S. Tjiang.: Code Generation Using Tree Matching and Dynamic Programming. ACM Transactions on Programming Languages and Systems, **vol. 11, No. 4**, (1989)
2. G. Arocena, A. Mendelzon, G. Mihaila.: Applications of a Web Query language. In Proceedings of the 6th International WWW Conference, Santa Clara, California, (1997)
3. G. Arocena, A. Mendelzon.: WebOQL: Restructuring Documents, Databases, and Webs. In Proceeding of the SIGMOD Conference, Seattle, (1998)
4. N. Ashish and C. Knoblock.: Wrapper Generation for Semi-structured Internet Sources. In ACM SIGMOD Record **vol.20 No.4**, (1999)
5. B. S. Baker.: Parameterized Pattern Matching:Algorithms and Applications. Journal Computer and System Sciences, (1994)
6. J.A. Bergstra, P. Klint.: The Discrete Time ToolBus, In Science of Computer Programming, **31(2-3)**, (1998)
7. F. Ranno, S. K. Shrivastava, S. Wheater.: A Language for Specifying the Composition of Reliable Distributed Applications, Technical Report 17, Esprit LTR Project **No. 24962**, Project Report, C3DS Project. Dept. of Computing Science, University of Newcastle upon Tyne, (1999)
8. A. Gal, S. Kerr, J. Mylopoulos.: Information Services for the Web: Building and Maintaining Domain Models. International Journal of Cooperative Information Systems, **8(4)**, (1999)
9. L. Giu, C. Pu, W. Iian.: XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources. In Proceedings of ICDE'2000 (2000)
10. K. Kontogiannis, R. Gregory.: Customizable Integration in Web-enabled Environments, Lecture Notes in Computer Science, Engineering Distributed Objects (2001)
11. J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo.: Extracting Semisructured Information from the Web. In Proceedings of the Workshop on Management of Semistructured Data. Tucson, Arizona, (1997)
12. G. Huck, P. Fankhauser, K. Aberer, and E.J. Neuhold.: JEDI: Extracting and Synthesizing Information from the Web. In Proceedings of Cooperative Information Systems (COOPIS), New-York, (1998)
13. L. Liu, et.al.: CQ: A Personalized Update Monitoring Toolkit. In Proceedings of the ACM SIGMOD, (1998)
14. E. Myers, W. Miller.: Approximate Matching of Regular Expressions. In Bulletin of Mathematical Biology, **Vol.51 No.1**, (1989)
15. A. Sahuguet, F.Azavant.: Wysiwyg Web Wrapper Factory (W4F). In Proceedings of WWW Conference, (1999)
16. VoiceXML Forum.: VoiceXML Specification 1.0, 2000. In URL: http://www.voicexml.org/ (200)
17. WAP Forum.: Wireless Application Protocol Architecture Specification 1998. In URL: http://www.wapforum.org (1998)
18. World Wide Web Consortium (W3C).: Extensible Markup Language (XML) Version 1.0, (1998).
19. World Wide Web Consortium (W3C).: Extensible Markup Language (XSL) Version 1.0, (1998).
20. World Wide Web Consortium (W3C).: XML Path Language (XPath) Version 1.0, (1999).
21. Y. Zou, K. Kontogiannis, Web Based Specification and Integration of Legacy Services. In Proceedings of CASCON 2000, IBM Toronto Laboratories, (2000)

An Extensible, Human-Centric Framework That Promotes Universal Access to Electronic Commerce

Jacob Slonim, Theodore Chiasson, Carrie Gates, and Michael McAllister

Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada
{slonim, theo, gates, mcallist}@cs.dal.ca

Abstract. The traditional, evolutionary, computing-centric approach to software development has yielded computerized systems that are inaccessible to a significant percentage of the population due to the complexity associated with their use. To increase the accessibility of systems, we propose a human-centric approach to development that is based on personalization and personal ownership of private information. This paper outlines an extensible peer-to-peer framework in support of a human-centric environment that is configurable by domain experts, such as psychologists or physicians, who are not computer science or computer engineering professionals. The aim of this paper is to provide a description of the high-level architecture of this new human-centric framework.

1 Introduction and Motivation

As computerized systems continue to permeate every aspect of our society, a paradigm shift is occurring in the type of end user accessing these systems [6]. Historically, the end users of systems were highly trained professionals, and system development was heavily constrained by the limitations of memory, cpu cycles, and other resources. Much of the complexity in systems was therefore "pushed" out to the end users, since it was technically infeasible for the system to handle everything. Now that non-technical users are accessing computerized systems, a paradigm shift must occur from a computing-centric to a more human-centric approach [5,12,4].

During the past four decades, old hardware systems have been repeatedly replaced by newer, faster, and less expensive components, thus following a revolutionary pattern of advancement. This revolutionary pattern of advancement has been supported by the proliferation of personal home computers. Software systems and applications, on the other hand, have followed a more evolutionary path of advancement, where each new release maintains backwards compatibility with the earlier systems. In addition, the functionality offered by software systems has increased steadily, and hardware deficiencies were often overcome through software fixes. Software systems that have evolved over a long period of time still have some original components from as far back as thirty and forty years ago. For example, some operating systems still maintain 8-bit support,

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 116–126, 2001.

while the underlying systems are now based on 32 and 64-bit technology. This reflects a trend in software development where there is a great reluctance to delete code that is no longer needed. These factors have contributed to the ever-increasing complexity of software systems.

Development of software systems has traditionally been accomplished by software professionals for software professionals. The development group has evolved from a single "guru" into teams of programmers, but the domain knowledge of these teams is still technology-focused. Lack of knowledge in the application-specific domains has resulted in a computing-centric environment that end users find unwieldy to operate.

A dramatic increase in the number of computer users, high-speed network connectivity and lower costs have all contributed to a societal shift towards a knowledge-based economy [7]. The number of end users has risen so dramatically that a revolutionary approach to software development is becoming more feasible. There are sufficient numbers of new users to justify a market for domain-specific, specialized devices that do not follow an evolutionary pattern of development. As an example, the Waterloo-based company RIM has recently introduced a device that is only capable of sending and receiving email [3]. The demand for this type of product is based in part on the number of non-technical end users of systems who are faced with an overly complex model of interaction when dealing with computerized systems.

While advances have historically occurred from a "technology push" stance that assumes a certain capability set in the end user, our research aims to develop a human-centric environment that will allow advances to occur from a "technology pull" stance with the direct participation of non-technical domain experts. This human-centric environment will be achieved by building on the three-layer (physical, operating system, middleware) model and introducing a new human-centric layer above the middleware [8]. This paper focuses on the overall theoretical framework of this new human-centric layer, and not on implementation issues.

The remainder of this paper is structured as follows. Section 2 introduces the assumptions and approach we are taking in this research project. In section 3, a high-level description of the proposed framework is presented, and section 4 concludes the paper.

2 Assumptions and Approach

The main goal of this research project is to achieve universal accessibility to applications such as electronic commerce. The approach we take is to develop a prototype of a human-centric framework that will allow domain experts to configure the system, allowing for a less computing-centric approach to the evolution of systems. To this end, the research defines a framework that bridges the gap between domain experts who can perform comprehensive analyses of tasks and computer systems that fulfill the tasks. Personalization information is the

key to allowing dynamic customization of interactions with end-users in order to increase accessibility to applications.

This research project adopts a human-centric view of systems. We fundamentally extend the client-server model of interaction to the more general peer-to-peer model [2] to facilitate end-user ownership of data. Adopting a peer-to-peer approach allows the end-user system to act as a peer in the environment that can respond to queries from remote systems. Within an interaction between two users, each user must maintain autonomy to safeguard their data. As peers, the system can negotiate the mode, duration and context of a transaction to meet common requirements, whether for security, interaction features, network throughput, or some other criteria. The peer-to-peer architecture also allows the system to exploit the computational power of multiple devices rather than centralizing the processing. Since existing legacy systems and middleware follow the client-server paradigm, we propose a hybrid system that allows for peer-to-peer interactions at the human-centric layer but still relies on the client-server middleware to access legacy systems. In the discussion of the components of the human-centric layer in section 3, each component will support peer-to-peer interactions.

For the purposes of this research, it is assumed that users own their personal data. Ownership of personal data is closely tied with user privacy; users should be allowed to choose with whom they share their information, under what circumstances the information is shared, and how long another person or company can retain the information after it is released. These exchanges may require a combination of technical functionality and policy. For example, a certificate can validate data over a specific period of time, but policy mechanisms are required to ensure that stale information is discarded once the certificate has expired. The security component of the human-centric layer will provide mechanisms for users to manage the privacy of their data by relying on reasonable default and minimum security schemes for data as recommended by domain experts or legal policy, by managing access to and from the data, and by negotiating secure methods of exchanging data.

Another premise of the current research is that personal electronic computing devices will become common. These devices include personal data assistants (PDAs), organizers, cellular phones, and wearable computers. Our research aims at increasing accessibility through these devices, or through combinations of these devices, by using the best features of each device in solving a single task. The computational and memory capacities of PDAs have consistently been about four years behind that of personal computers. Consequently, tasks that involve office computers today may be reasonably targeted for personal devices within a few years.

This research project assumes that security should be configurable at the end user level, allowing for personalized security policy management. Security mechanisms such as biometric authentication and location-dependent context based on GPS systems should be supported within the human-centric framework.

The personalization component of the human-centric layer includes generic algorithms and sets of tools in support of machine learning to dynamically adapt personalization information over time. Planning tools will facilitate the optimization of tasks based on constraints in the personalization information and diverse information sources concerning availability, cost, and quality of product offerings. These planning tools will be configured by domain experts to adjust dynamic personalization information based on triggers set on domain specific profile views.

The interface component of the human-centric layer provides a framework for dynamic configuration of devices and drivers to facilitate personalized end-user interface mechanisms. Personalization information will dictate the appropriate interaction mechanisms for individual end users, and transformation drivers may be employed to convert content to appropriate formats during content delivery.

Ideally, businesses would allow access to their underlying data through SQL queries, and the human-centric layer could access data directly and process it locally to personalize interactions for the end-user. The human-centric layer will, however, support client-server interactions for accessing legacy servers that do not adopt the peer-to-peer model. When direct database queries are not possible, agents will interact with the legacy servers through their public interfaces (typically www pages) and parse the results of these pages for local processing.

In combination, the personalization, security, planning, and user interface components of the human-centric layer will provide a framework that allows domain experts to configure profiles and tasks from a "technology pull" stance. The personalization information will facilitate customization of end-user interactions, yielding a more accessible environment than the current computing-centric situation.

3 Framework

In this research, we build on the three-tier architecture that was introduced to resolve issues of heterogeneity in the client-server model. We introduce a fourth layer, called the human-centric layer, that uses a peer-to-peer model to support the personal ownership of information. Personalization is the key to increasing accessibility [9], but users must have full control over how their personal information will be shared before they will place it in the system. Using a peer-to-peer model allows for information to be stored securely on the user's devices and allows part or all of this information to be shared during interactions based on the user's privacy policies. The human-centric layer will provide a consistent, personalized interaction model to the end-user and will hide the complexity of the underlying interaction models required by the lower layers.

Interactions can be one-to-one, many-to-one, or many-to-many. One-to-one interactions represent transactions between a single user and a single business. In the case where a user wants a number of items from diverse sources, and they want either all of the items or none of the items, we have a many-to-one situation. The many-to-many situation arises when a community of users

cooperate to buy items from a number of suppliers in order to achieve lower overall costs by buying or shipping in bulk quantities. Each of these modes of interaction will be supported by the human-centric layer.

The human-centric layer will be modular and dynamically configurable. Some, but not all, of the core components include the personalization, security, planning (not discussed in this paper), and user interface components (see Figure 1). Additional components can be developed and configured in the human-centric layer to support new modes of interaction both from the end-user domain perspective and from the underlying system infrastructure perspective. Each component will provide sets of tools to domain experts that allow them to configure domain-specific personalized interactions. These configurations will rely heavily on personalization information to provide customized experiences to end-users.

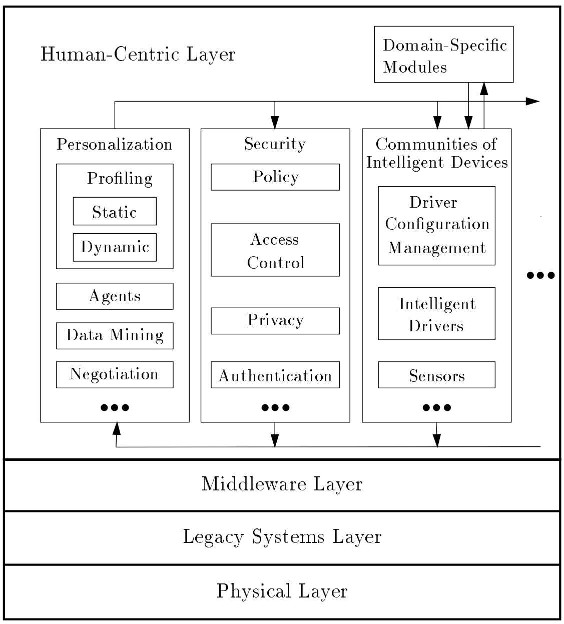

Fig. 1. Four layer model

3.1 Personalization Component

The personalization component is a dynamically configurable framework that includes agents and sets of tools for static and dynamic profiling, data mining, planning and negotiation. Personalization information will consist of sets of attribute-value pairs, constraints on allowable values, and links to other information sets. Placing the constraint definitions in the profiling information allows for

individualization of constraint settings at the end-user level. Constraint definition tools will allow domain experts with no programming background to specify or modify constraints.

Static profiles will be represented as views on sets of personal information and links to dynamically generated content. Dynamic profile information will be stored in sets derived from static profile information, external sources, interactive user contributions, and results from planning activities. Triggers will activate dynamic profile updates and activate calls to planning and negotiation modules. As more knowledge of the end-user is accrued over time, interactions should become easier for the end-user since the system will be better able to accommodate the user's personal preferences.

A single end-user will have one or more distinct profiles. Examples include a food profile covering information on grocery shopping and tracking what is "in the cupboard", an entertainment profile tracking favorite restaurants and leisure activities, and a personal medical profile. These profiles can share the same underlying tables of personal information and can each set their own database triggers for actions to be performed when a given context is reached. A global user profile will also exist that integrates information from all of these profiles and performs global tasks such as budget estimation.

Storage of personalization information can lead to privacy problems if not properly protected. Information can be protected at different levels of granularity within the profiles. For example, medical profiles, while owned by individuals, can be accessed by doctors and hospitals if the user chooses to allow this. Different facts within a medical profile require different privacy levels. While a user might want to disclose to everyone that they are a diabetic in an emergency situation, they may not want all of their medical history to be exposed. Support for levels of granularity will be accomplished by establishing subset views, which can themselves have constraint and security rules applied to them. Establishing profiles in this way leads to personal ownership of personal information, with the ability to share that information to whatever level is desirable. The personalization component will use the tools in the security component to establish privacy policies on profiles. Security tools will enforce security policies and ensure authentication and validation before allowing access to personalization information. Privacy tools will ensure that unauthorized users, applications, devices, or software components cannot access personal information.

Template profiles can be developed for use in automatic profile generation in order to minimize the redundancy and maximize the consistency of the end-user profiles with the system. Experts in a specific field can develop default constraints and attribute values for profiles in their area. For example, nutrition and medical experts could collaborate on generating templates for end-user grocery profiles for persons with particular illnesses. Thus, it should be possible for someone with diabetes to instantly generate a grocery profile based on their own food preferences and budgetary constraints in conjunction with the default profile supplied by the experts. This automatically generated profile would be

customizable by the end-user. Constraints could be placed by experts to flag warnings where appropriate.

The interface configuration module will ensure that information is presented to end-users in a format they can understand based on their personalization information and their current computing environment (i.e., handheld mobile device vs. desktop computer).

3.2 Security Component

The security component of the human-centric layer includes sets of tools for policy establishment and maintenance integrated with the overall security of the system. This component will provide extensible mechanisms for specifying authentication and access control techniques and protocols. The policy module will use a rule-based system that governs the interactions between the user and any other party, and will enforce access control to profile information. The granularity of protection can be varied from a single attribute value through to entire tables of personalization information. This flexibility will support a wide variety of configurations for the protection of private information, the establishment of secure transmissions, and the authentication of end users.

The configuration of policy information can be quite complex, and this complexity must be hidden from the end user of the system. Tools will be supplied for domain experts to create default security profiles related to their domain without specific knowledge of the underlying security mechanisms. This default domain-specific security policy will be added to a user's rule-base when they first interact with a new domain. The rule-base interactions will ensure that the user's overall security preferences are adhered to in addition to any more restrictive constraints specified by the domain expert's default rule-base. Similarly, policy experts can describe minimal default rules based on jurisdictional issues such as country of residence. These rules can then represent the default rule-base for users who are resident in that jurisdiction. The personalization information can also include lists of blocked sites that can never be visited on the user's behalf.

The rule-base in the policy module will dictate the authorizations required for various transactions and interactions. The relative security and trust associated with each authentication method is defined within the security policy module. The security module will determine the appropriate level of security required with regards to the type of encryption to be used and the acceptable authentication and certification mechanisms for each of the parties involved in an interaction.

When authentication is required from the user, the security policy will be consulted to determine what mechanisms can be employed. In addition to the security policy, the user's personal profile will be consulted to determine the limitations of the user, the capabilities of the device currently being used, and any predefined site restrictions. Based on a combination of this information, an appropriate authentication method will be chosen by the security component.

In addition to the more standard methods of authentication (tokens, passwords and biometrics) the security policy module can also specify location as an

additional constraint on access control. Global positioning system (GPS) information can be used to determine a user's exact location or the user's proximity to another GPS-equipped user. For example, specific security policies can be activated when the user is in a bank or in the presence of a physician. These methods can provide additional authentication for data security. For example, a user can define that any financial transaction greater than a certain amount must be performed at a bank. GPS information can then be used in conjunction with mapping information to determine if the user is at a bank branch during the transaction. Another example of location providing security involves the user's medical data. In this case, the user's medical profile may be encrypted, and only be decrypted if the user is in the presence of a registered physician. Here, the user's personalization module will be negotiating with that of the physician. A GPS can determine the relative location of each party, ensuring that they are within a certain proximity of each other. Additionally, the user's security module will authenticate that the other party is a registered physician by verifying the certificates presented. In life-or-death situations, mechanisms can be employed to overcome privacy policies. As an example, a traveller's medical profile might be made available in the event the traveller enters a registered hospital.

Data mining techniques can provide additional security. Each transaction the user makes can be recorded and analyzed so that the user's profile also contains trends that describe the user's typical behaviours. An action can be triggered if the user performs an atypical transaction according to the profile developed to date. The actual action triggered will be defined by the security policy and will govern the access that is provided. The policy may be strict, specifying that any abnormal action cause the device in use to shutdown and no longer respond to the user, or the policy may be less restrictive, specifying only that the transaction be flagged as suspicious while still allowing access to the data through normal access control requirements.

Access control mechanisms are required to allow users access to their own information. There are several circumstances under which a user would need to authenticate themselves, such as when performing a bank transaction, visiting a doctor's office to have a prescription renewed, or reading email. Each of these requires a different level of security commensurate with the value or perceived value of the interaction. For example, a user may not require that email be particularly secure, yet want a high level of security protecting banking transactions. The user's security policy dictates the authentication required for each of these accesses.

3.3 User Interface Component

There is much current research on achieving universal accessibility through adaptive user interfaces. The Archimedes project at Stanford is developing the Total Access System [10] that provides a personalized interface through hardware devices. Abrams et al [1] approach the universal access problem through User Interface Markup Language (UIML)-based device independent interfaces. UIML allows a developer to write the interface once with a single description and use

style sheets for each individual device. Commercial UIML renderers can convert between UIML and many industry standard markup languages. As another alternative, Shim et al [11] focus on content equivalence in their Synchronized Multimedia Language (SMIL). SMIL specifies a timeline based synchronized multimedia presentation and enhances universal accessibility by allowing the provision of alternative equivalent content for audio, video, and images.

Traditionally, accessibility research has focused on the user interface. This approach can help to improve the "look and feel" of an application, but cannot address the underlying complexity of the system or the complexity of information overload. Our approach is to drive the interface design through analysis of the tasks in combination with the personalization information for each end user. This will allow us to build user interfaces with reduced functionality by tailoring the interface in a domain-specific way and handling as many details as possible automatically based on the personalization information.

The user interface component will manage sensors, context sensitive interaction drivers, and translation services that may be multi-modal. Proper management will draw on research from many areas that include universal accessibility, multi-device coordination, device independence, personalization, and multimodal interfaces. The challenge is to combine results from these areas under a coherent, complete, extensible, and transparent framework.

To achieve personalization, domain experts specify which aspects of a profile directly affect the task at hand and what modes of interaction are best suited to the task. User interface domain experts will specify equivalences between devices or interaction modes based on preferences from personalization information, and the system will tailor interactions with the end-user accordingly. The interface configuration module will be responsible for translating between communication modes based on personalization information and the current computing context. For example, translation from text to speech, speech to text, text to Braille, or language to language is fairly standard and can be implemented dynamically. The module will be dynamically configurable, so that new translation drivers can be added as they become available.

4 Future Directions

There are still some unresolved issues to address in the design of the detailed human-centric layer of the framework, including a detailed description of the planning component, the identification and description of other required components, and the inter-component interface descriptions. We are in the initial stage of developing a prototype of the human-centric framework based on commercially available products.

Personalization information sets will be represented as tables in a commercial database system. Derived sets, subsets, set unions, and set intersections will be represented as views on the base tables. Security restrictions will be implemented using database security mechanisms, which allow for course and fine-grained access restrictions. Domain expert configurations will be represented as workflows

using a commercial workflow product. Each step in the workflow will use available personalization information to facilitate ease-of-use for the end user. For example, if the user's name, address and phone number are required by the grocery shopping workflow, the user will not be prompted for input since this data is readily available from their personalization information. Exception handling within the workflow will be configurable by domain experts to request missing information from the user or from external data sources.

Configuring these commercial products with a human-centric focus allows for a dynamically configurable and extensible framework geared to increasing the accessibility of systems for all end-users.

5 Conclusions

Our approach to increasing universal accessibility is through personalization of interactions with end-users, given that end-users own and control their own data. We introduce a human-centric layer that relies on the peer-to-peer paradigm. This layer provides a framework that allows domain experts to apply their domain knowledge in the configuration of personalization information and task descriptions without the intervention of computer science or computer engineering professionals.

Rather than attempt to solve the problem of accessibility through the traditional "technology push" approach, this research describes a framework in which domain experts can apply their specific domain knowledge to configure domain-dependent personalization characteristics and tasks, thus encouraging a "technology pull" approach. End users of systems will then use minimally featured devices or interfaces to achieve domain-specific goals, where any subset of the functionality that can be handled internally will be done automatically based on personalization information. The use of static and dynamic profiling will allow the system to learn a user's personalization information over time. Re-purposing of existing commercial product offerings that have historically been employed for business enterprise computing will allow for a rapid prototype implementation of the human-centric framework.

Acknowledgements. This research is supported in part by grants from the IBM Canada Center for Advanced Studies, the Izaak Walton Killam Fund for Advanced Studies at Dalhousie University, and the National Sciences and Engineering Research Council of Canada.

References

1. Abrams, M., Phanouriou, C., Batongbacal, A.L., Williams, S., Shuster, J.E.: UIML: An Application-Independent XML User Interface Language, 8th International WWW Conference, 1999

2. Bauer, M.A., Coburn, N., Erickson, D.L., Finnigan, O.J., Hong, J.W., Larson, P.-A., Pachl, J., Slonim, J., Taylor, D.J., Teorey, T.J.: A distributed system architecture for a distributed application environment, IBM Systems Journal, Vol. 33, No. 3, 1994, pp. 399–425
3. Blackberry: RIM Delivers Wearable Wireless Device Based on Embedded Intel Architecture, Press release, http://www.blackberry.net/news/press/1999/pr-19_01_1999-02.shtml, 1999
4. Chiasson, T., Gates, C.: Electronic Commerce Universal Access Device – The Knowledge-Acquiring Layered Infrastructure (KALI) Project, ACM Crossroads, Nov. 2000
5. Chiasson, T., Slonim, J.: Mediating between Computing-Centric Systems and Human-Centric Interfaces, Proceedings of the International Conference on Telecommunications and Electronic Commerce, 2000
6. Computer Science and Telecommunications Board, NRC, Fostering Research on the Economic and Social Impacts of Information Technology, National Academy Press, 1998, Washington, D.C.
7. Dickenson, P., Ellison, J.: Getting connected or staying unplugged: The growing use of computer communications services. Services Indicators, 1st Quarter (1999)
8. Emmerich, W.: Software Engineering and Middleware: A Roadmap, Proceedings of the 22nd International Conference on The Future of Software Engineering 2000, pp. 117–129
9. Kramer, J., Noronha, S., Vergo, J.: A User-Centered Design Approach to Personalization, Communications of the ACM, Vol. 42, No. 8, 2000, pp. 45–48
10. Scott, N.G., Galan, J.B.: The Total Access System, CSUN 1998 Papers, 1998
11. Shim, S.S.Y., Gao, J.Z., Wang, Y.: Multimedia Presentation Components in E-commerce, Advanced Issue of E-Commerce and Web-Based Information Systems, WECWIS 2000, pp. 158–165
12. Slonim, J., Chiasson, T., Gates, C.: Creating an Electronic Commerce Device which Promotes Universal Access: The KALI Project, Lecture Notes in Artificial Intelligence, 2000, Vol. 1830, pp. 2–12

CBR-Responder, an Automated Customer Service for E-Commerce

Yao Hui Lei, Gang Mai, and Esma Aïmeur

Université de Montréal
Départment d'informatique et recherche opérationnelle
C.P. 6128, succ. Centre-Ville
Montréal, Québec, H3C 3J7 Canada
{leiyaohu, maig, aimeur}@iro.umontreal.ca
http://www.iro.umontreal.ca/

Abstract. There is increasingly intelligent client support in electronic shops. Its main purpose is to help the customer search the products quickly and efficiently. Usually, customers need product service from the electronic shops. Therefore, customer service is very important to the company. Many companies have made efforts to provide customer service systems or eService systems on the Internet. Unfortunately, these eService systems always need interaction with a (human) customer service representative (CSR). The automated customer service is poorly provided if at all.

Case-Based Reasoning (CBR) is a recent approach to problem solving and learning that has been applied to various domains. In this paper, we introduce CBR to improve on customer service and we propose a direction to solve the problem. We also implement an application, which we call CBR-Responder, to illustrate how it works. We use CBR and Bayes classification technology to respond automatically to the comments of customers.

1 Introduction

The objectives of E-commerce involve the increasing speed and efficiency of business transactions and processes. As the Internet industry grows, so does the need for improved customer service [10] on the Internet. The customer service in a sales system is an important section of a company. However, pre-sale and post-sale service could not keep up with the rapid growth of e-commerce. It is a key problem to deal with customer comments efficiently.

Traditionally, the company has to hire many *Customer Service Representatives* (CSR) to respond the comments manually. On one hand, the company pays their salary; on the other hand, these CSRs have to be trained in order to cope with the problems. It has been surveyed [11] that support calls cost \$33 each by telephone, \$10 by email, and \$1.17 for a web-based search.

E-companies are making efforts to take advantage of the Internet in order to reduce customer service costs. This problem has been solved to some extent.

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 127–137, 2001.

Currently, there are several types of methods available to help E-companies deal with customer service interaction. These methods are called *eService* [13]. For example, the customer has a live talk with a CSR over the web. The company provides an eService system to deal with incoming e-mails. These systems require 100% human intervention. Usually, they are the principal parts of the phone-based customer service technology on the Internet.

In fact, we found that most of the customer comments could be divided into some categories to which we could give the same response. In other words, it is possible to introduce an automated mechanism to deal with these comments.

This paper discusses different ways to implement customer service. Section 2 describes some eService systems used on the Internet and gives the weak part of these systems. We focus on the *IntelliServe* system [6,1] to analyse the classification algorithms. Then we give our approach with CBR technology in Section 3. Section 4 describes the application *CBR-Responder* that we have developed. Finally, we compare our approach with the IntelliServe system and discuss future work.

2 Current Approach

There are some *web-based eService* systems found on the Internet. *Servicesoft* [12] integrates self-service applications, e-mail management and live collaboration, which are all based upon a sophisticated knowledge base that delivers the consistent information and advice required to accurately answer customer (and employee) inquiries.

The *icommunicate* system [8] is the software of the evolution of multi-channel, online customer service. It offers customers and support staff the benefits of combined self-service/self-help and live/personalized customer care. It provides a sophisticated customer retention system.

Another subsequent system based on message classification, IntelliServe, is developed by Navator company [1,6]. It uses the *Bayes classification* [7] and *regular expression* [1,6] algorithms to classify the customer's comments. This system implements automated customer service to some degree.

All these systems require the company to build a sophisticated application network and a sophisticated knowledge base. They also require experts to work with the system at the same time. Consequently, the automation of these eService systems is fairly poor.

After a glance at the IntelliServe system, we focus on how we can give a better performance to implement an automated eService.

2.1 Description of IntelliServe

IntelliServe focuses on providing the customers with *immediate, relevant, consistent*, and *accurate* responses to *frequently asked questions* (FAQ). It can respond to up to 80% of customer questions and comments — with a less than 3% error

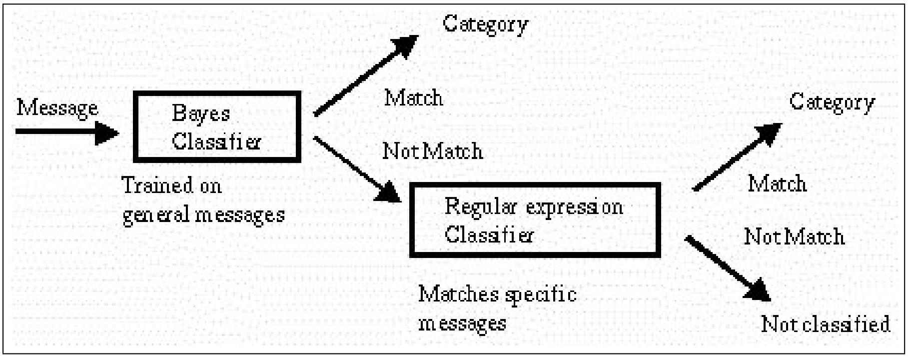

Fig. 1. Message classification in IntelliServe system

rate — without the intervention of a CSR. In the case that IntelliServe cannot provide a response, the software can route the comment to a CSR for resolution.

The data have shown that the majority of customer questions and comments fall into the FAQ category. The system provides immediate and automatic responses to these comments before the customer leaves the web site. Furthermore, it frees the CSRs from the burden of dealing with tedious questions and comments, allowing them to concentrate on those that require human intervention.

2.2 Workflow of IntelliServe

When a customer's e-mail is received (see Figure 1), it is classified by Bayes classifier and regular expression classifier. The Bayes classifier builds a probabilistic model for each message category. Each probabilistic model identifies which words are likely to be presented in messages of that category. The classifier must be trained on a large collection of messages previously classified manually. It is well adapted for general comments that can contain a wide variety of words.

The regular expression classifier is mainly about information extraction, which isolates the critical information from the message. The regular expression classifier is more complicated since it takes into account word order, word combinations, synonyms, punctuation, etc. It concerns the semantic of the message. To decide in which category does a given message fall into, the system has to analyse its semantic logic.

2.3 Disadvantages

Obviously, the regular expression classifier works poorly in practice because it has to set up a huge semantic database and it needs a communication with the company database and the conversation rules. Moreover, this system does not support machine learning. After a new comment has been solved by the CSR, the system does not put the solution into the predefined reply category. Therefore,

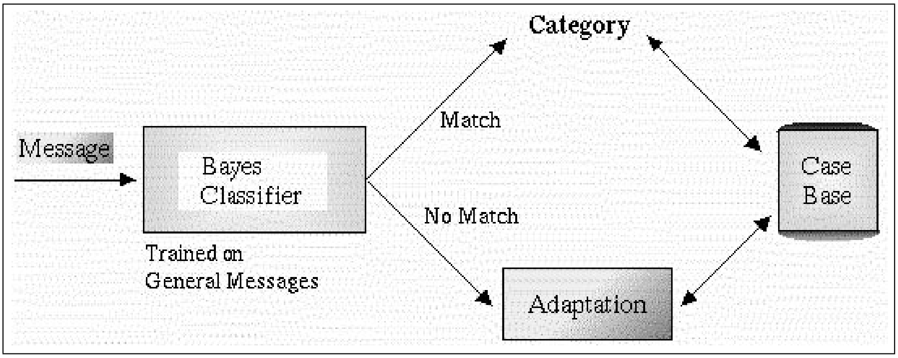

Fig. 2. Message classification with the CBR technology

when the same comment comes up again, the information goes through the classifier again and is sent to the CSR.

In the section below, we give our new approach based on the CBR technology. It is an alternative method, which is used instead of the regular expression classifier.

3 CBR Solution

During recent years, the *Case-Based Reasoning* (CBR) [4,5] has been applied in several domains of e-commerce [3]. Case-based reasoning is a problem-solving paradigm that in many respects is fundamentally different from other major AI approaches. Instead of relying solely on general knowledge of a problem domain, CBR is able to utilize the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. A second important difference is that CBR is also an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems.

3.1 Classification and CBR

As for the automated customer service system, we can see that it has the features of classification and similarity. It is necessary to have a case-base in order to store all the cases that are relevant to the domain under consideration. More interestingly, we need a technique to determine to which predefined category a given case belongs, and to find the most similar previously encountered cases. For this purpose, we use a Bayes classification technique (See Figure 2).

In this architecture, we replace the regular expression classifier with the important step of CBR technology: *adaptation*. The message is classified by the method of Bayes classification. We combine it with CBR technology in order to deal with the cases whenever they are classified successfully or not.

To find a *best-matching* solution, the system looks up the attributes of each case and calculates the maximum value of the probability, then it sends the response automatically to the customers via e-mail. Otherwise, there are no cases in the case-base that can match the new case, so this new case should be sent to a CSR for adaptation. The CSR solves the new case and puts it into case base in order to make the case base fit with this kind of case in the future.

3.2 Implementation Algorithm

Bayes classification is the traditional way to classify text. To give an estimated value for an attribute, the classification can be performed on test documents. It must be trained on a large collection of messages previously classified.

We have examined about 700 training comments (for example: *I have not received the product I ordered ten days ago. I think it is not one hundred percent guaranteed.*) from the clients and found there were about 70% of the cases that talk about the five categories: $C_1, C_2, \ldots, C_5$. We named these categories: *confirmDelivery, account, credit, groupDiscount, lowerPrice.*

From the training data, we obtained the probability value for each category (See Figure 3). For each category there are some attributes (keywords from comments). We define these attributes in each category of the comments and get categories C_1: Wc_1, Wc_2, Wc_3; C_2: Wc_1, Wc_2; C_3: Wc_1, Wc_2, Wc_3; C_4: Wc_1, Wc_2, Wc_3; C_5: Wc_1, Wc_2, Wc_3, Wc_4. We also got the probability for each word under the category, which is the probability that the word happens in this category. We call them $P(Wc_1 \mid C_1)$, $P(Wc_2 \mid C_1)\ldots$

To calculate the probability of the new case on a given category, we use Bayes classification formula:

$$PBC(i) = P(C_i) \times \Sigma(P(Wc_{ij} \mid C_i))$$

where, $PBC(i)$ is the probability of the message on the category i. $P(C_i)$ is the probability of category i that we have obtained from the experiment. $P(Wc_{ij} \mid C_i)$ is the probability of keyword j under category i. For each word of the message, if it is in category i, we add its probability($P(Wc_{ij})$) together. Then, we multiply the result with the probability of category(PC_i). The final result($PBC(i)$) is the probability of the message under category i.

We do this kind of calculation for each category (see Figure 3) until we have all the PBC values. The category which has the largest probability will be the corresponding category of the new case.

The procedure that describes the solution is given in Figure 4.

Example. The customer's comment is: *Would you please give me a lower price for the rose? I think it is too expensive.*

The procedure takes the first word of this comment: *would*. Since we had not defined this word as a keyword, the procedure takes the next word. This process continues until a keyword, *lower*, is recognized. When it takes the word *lower*, it adds the probability, *20*, of this keyword and the corresponding category *visited* and continues until all the words have been examined.

Category Name	PBCi	PBi	Wci	p(Wci | ci)
confirmDelivery	0	15	confirm	30
			delivery	30
			confirmation	30
account	0	12	access	30
			account	40
credit	0	9	credit	30
			account	40
			number	28
groupDiscount	150	15	group	30
			discount	30
			price	10
lowerPrice	1500	20	prices	35
			price	35
			lower	20
			expensive	20

Fig. 3. Categories and Attributes

```
PROCEDURE match(string NEWCASE)
BEGIN
    AWORD       : string;
    VISITED     : boolean;
    PBMAX       : integer;
    PBC         : integer;

    Take the first word of NEWCASE into variable AWORD;
WHILE AWORD ≠ NULL DO
        IF AWORD is a keyword of a category THEN
            Set this category and the keyword VISITED;
        END IF

        Take next word of  NEWCASE  into variable AWORD;
    END WHILE

    PBMAX := 0;
    FOR each category VISITED DO
        Calculate the PBC of this category according the
        keywords and the probabilities.

        IF PBC > PBMAX THEN
            PBMAX := PBC;
        END IF
    NEXT

    IF PBMAX = 0 THEN        /* Cannot find a solution */
        Send the NEWCASE into adaptation casebase;
    ELSE
        RETURN the solution;      /* find a solution */
    END IF
END PROCEDURE
```

Fig. 4. Matching algorithm

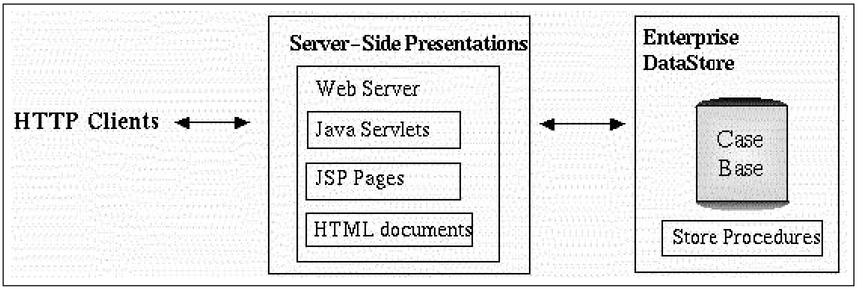

Fig. 5. Network architecture of CBR-Responder

Now, the procedure calculates the *visited* categories by adding the probabilities of the keywords (*lower, price, expensive*) together, then multiplies the probability of this category. We get PBC:

$$PBC = PB(category) \times (PB(lower) + PB(price) + PB(expensive)).$$

The value of PBC is displayed in the second column of the table.

The procedure checks other *visited* categories if they exist. If the value of PBC is bigger than the previous one, the new one is kept and the old one is discarded.

Finally, we select the largest PBC value. It is the best-match category of the customer comment. As we have calculated in the table, the category that has the largest PBC value is the *lowerPrice* category. Therefore, we send the solution of this category to the customer (not displayed in the table): *We appreciate your comments. We make every effort to keep our online prices competitive. In a recent review, we found that our prices were less than or equal to those of our major competitors on the Web.*

4 CBR-Responder

We have implemented an automated eService application on the Web. We call it *CBR-Responder.*

As we have discussed above, CBR-Responder uses the Bayes classification method to classify and match the customer's comments. The probability of the category and the probabilities of the words under the condition of the category are given according to the training documents.

4.1 Network Architecture

Since the customers send comments from their browsers, we utilize a client/server model to implement our web-based application (See Figure 5).

In this architecture, CBR-Responder gets the comments from the customer's browser and sends this request to the server-side application. The server-side

Fig. 6. Customer Interface of CBR-Responder

application is developed with Java servlets and JSP pages. It searches the corresponding solution through store procedures running in the enterprise datastore. The store procedures analyse the comments and look for the attributes from the case-base. The result probabilities are calculated by Bayes classification algorithm and the largest one is the best-match category.

When the best-match category is found, the corresponding solution or response is sent to the client via an e-mail address that is requested while he sends his comments.

In order to implement the adaptation step, we should add some operations to maintain the case-base. We provide several operations in our application: solve a case, add a new case, modify an existing case and delete a case. We give a glance of these operations in the following user-interface section.

4.2 User-Interface

As we have discussed above, we provide a customer interface to get the comments and CSR interface to maintain the case-base.

The customer interface is given below (See Figure 6). We give our example based on selling flowers in a FTD company [9]. The customer comments are related to flowers. The customer must provide his or her name and the e-mail address so that CBR-Responder can send the best-match response to him or her via e-mail as soon as possible.

After the user submits the comments, CBR-Responder looks for the best-match solution from the case-base, as we have calculated in subsection 3.2.

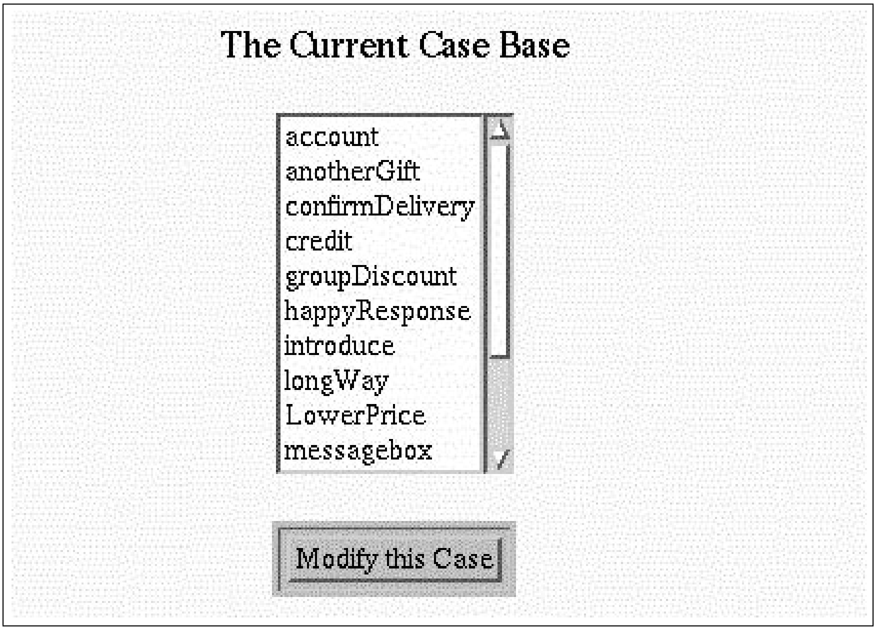

Fig. 7. Select a case from the case-base

The next step is the adaptation of the CBR. Since the case is new, CBR-Responder adds personalized information into the old solution. In this example, CBR-Responder adds the name (*Natalie Simard*) of the customer, so the final solution to the customer is:

> *Dear Natalie Simard,*
>
> *We appreciate your comments. We make every effort to keep our online prices competitive. In a recent review, we found that our prices were less than or equal to those of our major competitors on the Web.*
>
> *Sincerely yours,*
> *CBR Response Center*

Otherwise, the new case is sent to the unsolved case-base. The CSR will solve the new case manually and modify the case-base in order that CBR-Responder matches this kind of case successfully in the future. To implement this operation, the CSR selects a case that must be modified from the case-base (see Figure 7).

CBR-Responder displays the attributes and the solution of this case. The CSR can modify them and update the case (see Figure 8).

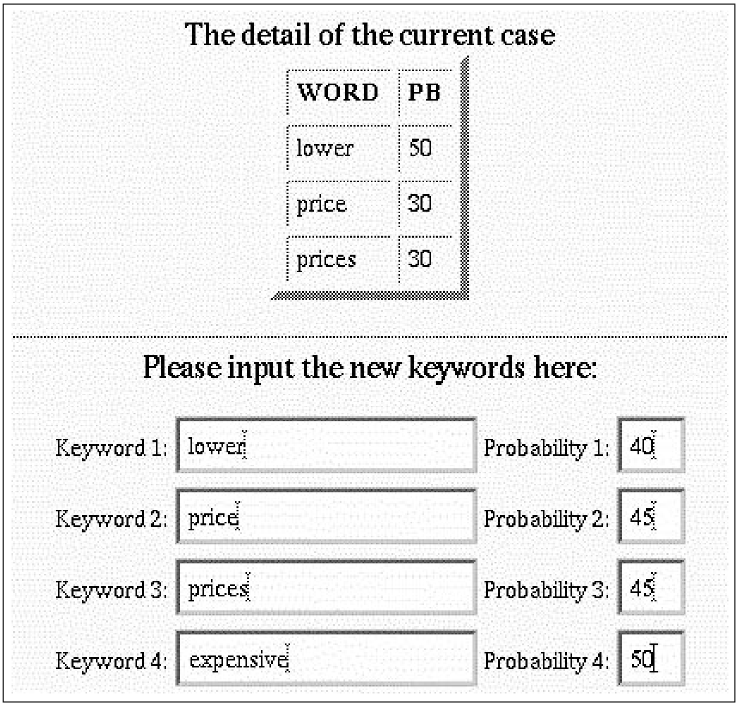

Fig. 8. Modification of the case

5 Conclusion and Future Work

In this paper, we raised the problem of the eService system on the Internet. We were motivated to use new technologies to address this problem because of the bad performance of current systems. We discussed ways to solve the problem and implemented an application system based on the web with Bayes classification and the Case-Based Reasoning (CBR) technology.

Our new system uses a Bayes classifier to deal with customer comments. If there has already been a solution for this case in the case-base, our CBR-Responder system responds without involving a (human) Customer Service Representative (CSR). On the other hand, if there are no solutions in the case-base, our system asks a CSR to solve the case, in contrast with the regular expression classifier of IntelliServe system. This function of our system is more powerful and more practical. Furthermore, it is easy to implement and maintain, including the need to support update, modify, delete and add operations.

Since adaptation is the important step of CBR, it is also very important in our system. Through adaptation, we add the new solution in the case-base, so that the CBR-Responder has the ability of learning. The more cases are encountered, the more powerful the system becomes, and the fewer interactions with the CSR are needed.

Our system had been put on the Internet and tested on various situations. We have received about 300 comments and we found that our system can deal with 30% of them. We believed that our system will become increasingly intelligent and powerful as it is trained on more customer comments.

As for future work, we intend to add a threshold to the case category. When the calculated maximum probability does not exceed this threshold, we fail to find the best-match solution. This case has to be sent to the CSR who is responsible for modifying the case-base in order to solve this kind of case automatically in the future. Furthermore, although we have tested our system on a reasonably large amount of data, which was sufficient to improve significantly the performance, the system should be trained on more comments so that we can make it more complete and powerful.

References

1. Lallement, Y., Fox, S. M., "Interact: A staged Approach to Customer Service Automation", Hamilton, H. J. (Eds.), Advances in Artificial Intelligence, AI-2000, Montréal, Canada, Lecture Notes in Artificial Intelligence 1822, Springer, pp. 164–175, 2000.
2. Bunke, H., Messmer, B. T., "Similarity measures for structured representations", in Richter, M. M., Wess, S., Althoff, K. D., Maurer, F. (Eds.), Proceedings EWCBR-93, First European Workshop on Case-Based Reasoning, University of Kaiserslautern, pp. 26–31, Nov, 1993.
3. Aïmeur, E., Vézeau, M., "Short-Term Profiling for a Case-Based Reasoning Recommendation System", Enric, P. (Eds.), Lecture Notes in Artificial Intelligence 1810, Machine Learning: ECML 2000, Ramon Lopez de Mantaras, pp. 23–30, 2000.
4. Watson, I., "Applying Case-Based Reasoning: techniques for enterprise system", San Francisco, Calif. Morgan Kaufmann press, 1997.
5. Kolodner, J., "Case-Based Reasoning", Morgan Kaufmann press, 1993.
6. Lallement, Y., Fox, S. M., "IntelliServe: Automating Customer Service", Artificial Intelligence for Electronic Commerce AAAI workshop. AAAI Press, page 1, 1999.
7. Kontkanen, P., Myllymaki, P., Silander, T., Tirri, H., "BAYDA: Software for Bayesian Classification and FeatureSelection", Agrawal, R., Stolorz, P., Piatetsky, S. G. (Eds.), Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98), pp. 24–28, AAAI Press, Menlo Park, CA, 1998.
8. http://www.icommunicate.com
9. http://www.ftd.com
10. http://www.brightware.com/news/press
11. http://www.thestandard.com/
12. http://www.livecontact.com/
13. http://www.magic-sw.com/

Introducing QoS to Electronic Commerce Applications

Gregor v. Bochmann[1], Brigitte Kerhervé[2], Hanan Lutfiyya[3], Mohamed-Vall M. Salem[4] and Haiwei Ye[4]

[1] University of Ottawa
[2] University of Quebec at Montreal
[3] University of Western Ontario
[4] University of Montreal

Abstract. Business to consumer is expected to be one of the fastest growing segments of electronic commerce. One important and challenging problem in such context, is the satisfaction of user expectations about the Quality of Service (QoS) provided when applications are deployed on a large scale. In this paper, we will examine the use of dynamic QoS management techniques in combination with replication at the various architectural levels of an electronic commerce application.

1. Introduction

It is expected that business to consumer commerce as well as other forms of electronic commerce will grow at a breakneck pace during the next four years. The value of goods and services traded between companies is expected to skyrocket from \$8 billion (U.S.) this year to \$327 billion (U.S.) in 2002, according to Sizing Inter-company Commerce, the inaugural report from Forrester Research's Business Trade & Technology Strategies service.

The business to consumer is expected to be the fastest growing segment of electronic commerce. However, there are several impediments that may have an affect on this growth. The most commonly cited problem is that of the lack of trust between businesses and consumers. Thus there is much research in security and payment protocols. Secure payment is only one aspect of concern about doing business on-line. Another, but less discussed problem, is user expectations about the Quality of Service (QoS) provided. By QoS, we are referring to non-functional requirements such as performance, availability and cost of using computing resources. For example, a user is bound to worry if there is a long wait during credit card processing. The user may even perceive the long wait to be a security problem. The time it takes for credit card processing should not exceed a certain threshold where that threshold is determined to be the average amount of time that a consumer is willing to wait on-line before hitting the "panic button". Satisfying the quality of service expectations of users for electronic commerce sites is becoming a considerable challenge that has not been addressed adequately.

QoS management refers to the allocation and deallocation of computing resources. Static QoS management techniques provide a guarantee that resources will be available when needed. Applying static QoS management techniques for electronic commerce sites typically involves adding additional resources e.g., adding additional

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 138–147, 2001.

servers or buying faster processors, faster bandwidth connections or more memory. It has been shown that this approach wastes approximately 20% of resource capacity and fails to satisfy quality of service needs in 90% of cases. This suggests that just adding resources is inadequate. In this paper, we will examine the use of dynamic QoS management techniques in combination with replicating web and application servers.

The outline of this paper is as follows. In Section 2 we introduce some typical Web server architectures in order to accommodate a large number of users. We also review the basic QoS parameters, which describe the user's perception of the provided service and the internal performance parameters of the distributed system components. We also consider differentiated service for different classes of users and some policies for managing QoS in this context. We note the importance of monitoring the actual QoS parameters while the system is running in order to provide for dynamic QoS management which automatically adapts to unforeseen situations, such as unexpected bottlenecks, partial system failures or unexpected user demands. In Section 3, we consider system architecture with a single Web server and e-commerce application and partially replicated database servers. We discuss the issues related QoS-aware distributed query processing which is feasible in this context. We consider different optimization criteria that may be related to different user preferences or the trade-off with the preferences of the system administrator. In particular, the network QoS parameters, available throughput and transmission delay, are taken into account. In Section 4, another replicated architecture is considered where the whole e-commerce application are replicated and a so-called broker process distributes the user requests between servers according to some QoS policies. Because of space limitations, only an overview of this approach is given. Section 5 contains the conclusions.

2. Systems Architectures and QoS Parameters

2.1. Basic System Architectures

The simplest form of an electronic commerce application as shown in Figure 1 consists of a web server as the interface for clients, an application server that has the program logic needed for implementation and a database server needed for storage of information.

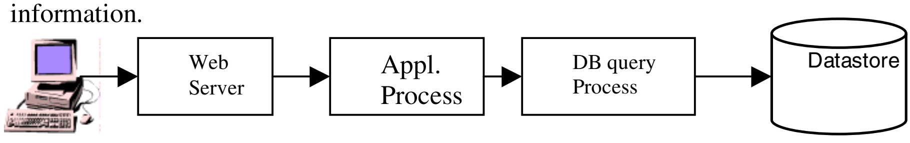

Fig. 1. Simple form of an electronic commerce application

A more complex electronic commerce application has its information reside on databases at different sites. For example, a virtual mall brings together services and inventories of various vendors and allows users to navigate through these vendors, adding items into a virtual shopping cart. The catalogue of services and inventories will often be on the databases at the vendor's site. If the information to be accessed is standardized, the DB query process can provides a uniform interface to the application process that hides the differences of the various catalogs and their access

differences. This would allow the straightforward processing of a query involving several vendors of the same mall, such as the following example query: *Find all sofas which price is less than $1000 and where a matching loveseat, recliner and coffee table can also be found.*

A typical architecture with a distributed data store is shown in Figure 2. Such distribution accommodates the distributed responsibility for the data, which is shared among the different vendors within the shopping mall. The distribution also increases the overall processing capacity since the different local query processes will run in on separate computers with their own data store; this will therefore increase the overall query processing capacity. The problem of QoS optimization in this context will be discussed in more details in Section 3.

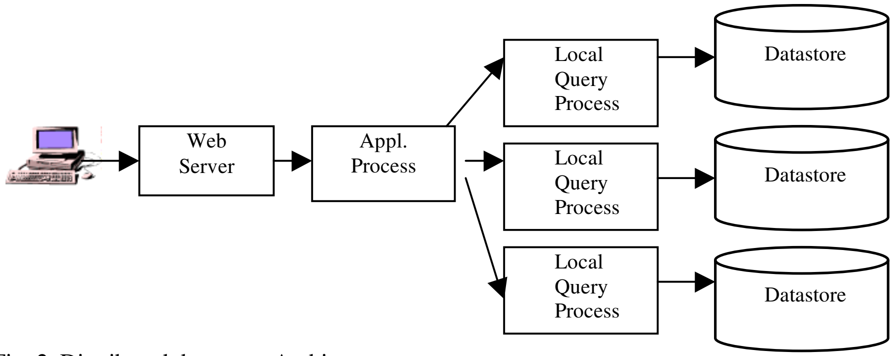

Fig. 2. Distributed data store Architecture

In the case of a very large number of users, a single server is not powerful enough to run the Web server and the application processes for all the users. In order to increase the processing power, one may distribute the Web server, application processes and query processes over different computers. Or one may go one step further and duplicate the whole system architecture, introducing a certain number of independent systems, all providing the same service, as shown in Figure 3. In order to distribute the user requests to different servers, we consider the introduction of a load distribution engine, which we call broker. The broker receives the initial user requests and determines, based on performance management information received from the Web servers, which server should be allocated to serve the given user request and the following interactions occurring within this new shopping session. Some of the issues related to the load sharing and QoS management for this replicated architecture will be discussed in Section 4.

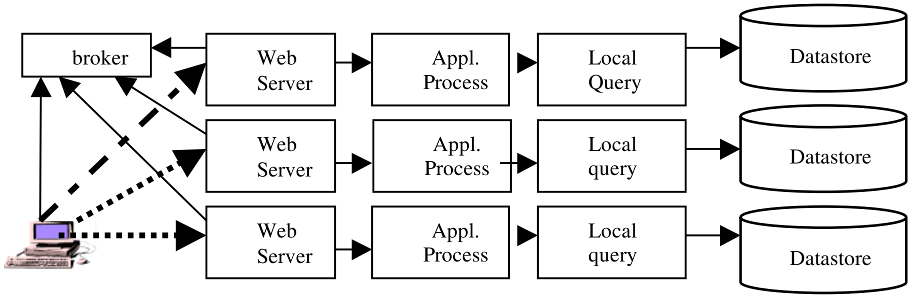

Fig. 3. Full replicated architecture

2.2. Quality of Service Parameters

With respect to performance, a primary user expectation about the QoS is specified in terms of the response time. The response time is affected by the transmission delay of the access network through which the client's workstation accesses the Web server, by the processing times in the Web server, the application process and the DB query process, and possibly the network latency between the Web server, application and database server if they are implemented on separate computers. In order to discuss QoS management in this context, we have to distinguish between the high-level QoS parameters relevant at the user level and the internal QoS parameters pertaining to the different system resources, which are used by the application. The goal of QoS management is to assure the controlled sharing of these resources among the different users according to certain management policies.

2.2.1. User-Level QoS Parameters

The following QoS parameters are important from the user's perspective:

1. Response time: This is the most important QoS parameter from the user's perspective. It is the time between the moment a request is sent to the time that the response has been provided to the user.
2. Availability: Availability is simply a measure of the system's effective "up-time". It represents the percentage of time the server is available during an observation period.
3. Servability: We call servability the percentage of time the server is available and can accept the user request. We assume that, in order to guarantee a certain quality of response time to the active users, the system will refuse new requests when the response time of the system exceeds a certain limit, which we call Rmax.
4. System throughput: This parameter is important from the perspective of the system owner, and it measures the number of user requests that are handled by the system. It is a measure of the amount of service that is provided. It is well known that the response time of a given system increases as the system throughput increases. When the maximum throughput of the system is attained, the response time becomes infinite since the internal queuing delays become arbitrary big. It is interesting to note that different query processing algorithms in distributed databases may lead to different maximum throughput and different response times (at less than maximum throughput). Therefore a compromise must be found

between maximizing the throughput and minimizing the response time. This is an important QoS policy decision to be made by the owner of the e-commerce system.

In the context of databases, the parameters of interest are database server availability, database management throughput and query response time. The throughput is usually measured in transactions per second [TPCW]. We consider the response time as a function of several parameters including the database server load (CPU usage, database connections, disk I/O activities), the network load (the available TCP throughput and delay), as well as the power of client machine.

2.2.2. Internal QoS Parameters

The internal QoS parameters pertain to the performance of the different system components within the considered architecture. They include the following parameters:

1. Network propagation delay: The time between the sending of a packet to its reception by the destination computer.
2. Network access capacity: This is the maximum throughput by which a given computer can send data over the network. It is determined by the network access link, which is relatively limiting in the case of modem access over telephone lines.
3. Effective network maximum throughput: This is the maximum throughput that can be effectively obtained between two computers over a given network. Even if the network access links allow for large throughputs, the effective throughput will be limited in most cases by the flow control mechanism of the TCP protocol since most Web applications and distributed databases use TCP for the transmission of data.
4. Response time of the query process; also its throughput (number of queries processed per second).
5. Processing delay and queuing time in Web server and application process; also their throughput.
6. Availability of the different servers and system components implemented on these servers.
7. Low-level performance parameters, such as CPU and memory utilization of the different servers that are part of the system architecture, as well as statistics related to data transmission over the network access links, etc.

2.3. Differentiated Classes of Users

We think that many future e-commerce systems will be able to provide different levels of service to different classes of users. For instance, the system may distinguish between a casual user (which is not known to the shop's organization) and a registered user who is a regular client of the shop. Some of the registered users may be known to buy many goods in the shop; they may obtain the "Elite" service, while the normal registered user obtains the "Premium" service and the casual user the "Normal" service. These different classes of service may differ in several aspects, such as the following:

- A higher class of service will have a shorter response time.
- A higher class of service may provide facilities, which are not available at the basic level, such as for instance a teleconference chat with a sales person.

- A higher class of service implies higher availability and servability.

2.4. QoS Policies

In the case of a single class of users, the general policy for QoS management may be the following:
Policy 1: At any given point in time, all active users should experience the same average response time. If the average response time reaches the value Rmax, the system should not accept any new user sessions. Note: Rmax may be infinite in the case that no user will be refused. It is clear that the size of the active user population (which varies over time), the system's processing capacity and Rmax determine the servability parameter of the provided service.

In the case of two classes of users A and B (where A is a higher class than B) there are different ways the priority of A over B may be specified. Assuming that all users of a given class should experience the same average response time and servability, and that the average response time for class A is smaller than for class B, we may consider the following policy alternatives:
Policy 2:

1. If the response time for class A reaches $Rmax_A$ then no new users of class A will be accepted.
2. If the response time for class B reaches $Rmax_B$ then no new users of class B will be accepted.
3. $Rmax_A < Rmax_B$

Policy 3:

1. If the response time for class A reaches $Rmax_A$ then no new users of class A nor class B will be accepted.
2. As above
3. As above

Note that policy (3) implies absolute priority of class A users over class B users. If there are enough class A users requesting service, no class B users may enter the system and eventually only class A users will be served. In the case of policy (2), the amount of processing capacity for each of the two classes of service is not specified and may be determined by the system administrator in an arbitrary manner, thus allowing in the case of general overload condition a somehow balanced number of class A and class B users. We note that giving higher CPU priority to the processing of the class A requests as compared with class B requests, may be used to implement both of these policies. However, the policies leave some freedom to the implementation about the amount of resources to be allocated to the different user classes, therefore strict priority may not always be appropriate.

2.5. Fault Management

In any dynamic approach to QoS management, a key problem is detecting why a QoS requirement is not being satisfied. For example, assume that under normal operating conditions the performance requirement for a Web server is as follows: "If the number of requests is less than 100 per second, the time it takes to service an HTTP request should be less than 1.5 seconds". If this requirement is not satisfied, this is a

symptom of a problem. The reason that this requirement may not be satisfied could be due to a number of problems including the following. The host machine that the Web server is running on has a high CPU load, the network between the Web server and the application / database server may be congested or the host that the database server is running on may have a high CPU load or be down. Determining that a QoS requirement is not satisfied is referred to as *fault detection.* Fault diagnosis techniques hypothesize as to a probable cause of the degradation, with subsequent testing used to determine the validity of the hypothesis.

Support of fault detection and diagnosis requires that the architecture supports the following: (1) Resource monitors for the network and the host machines that run the components of the application as well as monitors that measure application behavior that can be compared to the QoS requirements. (2) Management applications that can analyze the monitored information to determine if a QoS requirement has been violated and diagnose the cause.

2.6. Monitoring Application and Resource Behavior

The subsections above illustrate the need for performance monitoring. The broker of Figure 3, the global query process in Figure 2, and the management application described in Section 2.5 all need information about application and system run-time behavior. Thus, for application and system components we need an associated performance monitor. For example, a Web server would have instrumentation for monitoring its behavior and there would be a daemon process monitoring CPU load, incoming network traffic and outgoing network traffic for the computer on which the Web server is running. The monitored information used by the broker, query processor, and the management application all intersect, but are not necessarily the same. In addition, the frequency of the need for information will vary. This suggests that each process be able to register with a resource or application monitor for the timely and periodic sending of information. Alternatively, if a process does not want to register for timely and periodic information, the resource must provide an interface that allows for request of the monitored information when needed.

3. QoS Aware Distributed Query Processing

As shown in Figure 2, a typical real-world back-end database configuration for e-commerce systems consists of multiple database servers for providing stability, load balancing and better performance. In such configurations, the database content can be allocated to different database servers. For example, tables storing product catalog are less likely to be updated as compared to those tables used for order processing, thus these two types of tables can be distributed to different servers to get some performance gain. However, such kind of data distribution requires connecting the two tables when processing queries that need to access information from both tables. This requires the database management system (DBMS), or more specifically the query optimizer, to be aware of the data distribution. The new challenge, as compared to the context for traditional DBMS, consists here on how to take into account the dynamic nature of the underlying network and database server load. In addition, since users may have different expectations from the e-commerce server, while passing down to the database server, these expectations have to be mapped to different

optimization criteria. Therefore, the traditional query optimizer lacks the flexibility to configure to different optimization criteria. In this new context, we revisit distributed database query processing in the presence of information about the network quality of service (delay and available throughput) and user preferences concerning query optimization criteria [Ye99]. We consider such optimization criteria as minimizing the overall resource utilization, lowest response time or lowest cost (based on some tariff structure for network and database utilization) [Ye00]. We work in the context of federated and distributed databases communicating either over a local area network or over the Internet. Accordingly, our distributed query processing strategies can be described as "Quality of Service aware". The QoS aspect here refers to the dynamic nature of network, the server load and the user's preference.

Global query optimization is generally implemented in three steps [Daya85, Meng95, Ozsu99]. After parsing, a global query is first decomposed into query units (subqueries) such that the data needed by each subquery is available from a single local database. Second, an optimized query plan is generated based on the decomposition results. Finally, each subquery of the query plan is dispatched to the related local database server to be executed and the result for each subquery is collected to compute the final answer. In our study, we focus on the first two steps and map them to the problem of *global query decomposition*, *inter-site join ordering* and *join site selection* [Corn88, Du95, Evre97, Urha98].

Global query decomposition is usually complicated by duplication. Therefore this step is usually guided by heuristics. Two alternative heuristics [Ozsu99] can be employed during this step. The first alternative is to decompose a global query into the smallest possible subqueries and the second alternative is to decompose a global query into the largest possible subqueries. In our work, we assume the subqueries are generated based on the second heuristic. The reason is that we wish to push as much processing as possible to the component DBMS so that we could simplify the optimization at the global level and hopefully could reduce the data transmission among different sites. Therefore, the objectives of this step are 1) to reduce the number of subqueries and 2) to reduce the response time of each subquery. The first thing that needs to be done, when a query is given to the optimizer, is to split it into subqueries based on data distribution across multiple nodes. Thus the main task of global query decomposition is to decompose a global query into subqueries so that the tables involved in each subquery target on one local site.

Then in the join-ordering step, the optimizer tries to come up with a good ordering of how to combine those joins between subqueries. The join ordering can be represented as a binary tree, where leaf nodes are the base tables and internal nodes are join operations. Because we want to utilize the distributed nature of multi-database system, we try to make this tree as low as possible, which means we hope the join can be done in parallel as much as possible. In our work, we first build a left-deep tree using dynamic programming. And then, use some transformation rules to balance the linear tree to a bushy tree. In both steps, we consider both server performance and network performance captured by QoS monitor.

Last, we have to decide where to perform the inter-site join – this is referred as join site selection. In our approach, we implement this step by annotating the binary tree generated from join ordering step. Each node is annotated by a location of where this

join should be performed. The problem is how to integrate QoS information into these phases. Table 1 identifies the related QoS parameters factored into each step.

Global query optimization	Relevant QoS parameters
Decomposition	Server availability Server load
Inter-site join ordering	Server load TCP throughput Network delay
Join site selection	Server load TCP throughput Network delay

Table 1. QoS information relevant to global query optimization

We are presently working on enhancing these algorithms by considering the user preference to provide a differentiate service. For example, for higher priority user, the algorithm could generate a different plan by accessing different set of data nodes so that the result set of the query could have more interesting content such as multimedia data. We could also propose data distribution rules for locating data accessed by higher priority user to high performance database servers. In addition, the issue of how to corporate with other server replication, such as the one introduced in the following section, deserves a careful study.

4. Web Server Replication

A typical approach to improving response time is replication of a service. Replicating the database servers requires a good deal of synchronization of the copies of the data and thus it is not always feasible to replicate the databases. Replication of the web server and associated application servers do not have this overhead and thus this work focuses on web server replication. Client processes connect through a virtual server address to an intermediate host that forwards their requests to a replicated web server. The intermediate host is referred to as a *Broker* (see Figure 3). The *Broker* is used to direct Web traffic to one of a number of web servers. The selection is based on monitored information from the Web servers and administrative policies on how to use the monitored information to select a server. Examples of *administrative policies* include the following:

- **Balanced server performance.** Requests are assigned equally to the replicated Web servers in a round-robin manner or in a weighted manner based on the measured performance of the different servers. Different kinds of such policies are described and evaluated in [Salem 00].
- **Content.** Requests are assigned to Web servers based on the type of request. For example, a request for dynamic content pages may be assigned to one server and a request for static content may be assigned to another server. Another example is that video clips are retrieved from one server while text information is retrieved from another one. Yet another example is using client host information (e.g., host load, connection type) to determine the type of Web page that gets sent to the client.

- **User.** Requests are assigned to a web server based on the user class as discussed in Section 2.3. Users who have paid a fee may be designated as premium users and be assigned to their own server. If the server is in danger of being overloaded, then premium users may also be assigned to a second server, etc; while non-premium (i.e., non-paying users) get assigned to other servers.

5. Conclusion

The discussion in this paper shows that different architectures with server duplication may be adopted for e-commerce applications if the system is designed for a very large user population. In order to provide a service quality and allowing for different classes of users with different expectations, it is important to introduce dynamic QoS management for allocating available resources in the best possible manner. The QoS management decisions should be made dynamically based on measurements of the actually available service qualities from the network and the different server components of the distributed systems. This implies dynamic performance monitoring and making these measurements available to the QoS management processes.

Two distributed system architectures are considered in the paper. The case of distributed query processing in an architecture containing a single Web server and e-commerce application front-end together with a partially replicated database is considered. Another replicated architecture in which the whole e-commerce application (including Web server and database back-end) is replicated on different sites is only discussed without much detail. In our ongoing work, we evaluated the performance of different architectures and the optimization algorithms that can be used in these different contexts.

References

[Corn88] D. W. Cornell, P. S. Yu: Site Assignment for Relations and Join Operations in the Distributed Transaction Processing Environment. ICDE 1988: 100-108

[Daya85] U. Dayal, *Query Processing in a Multidatabase System*, In Query Processing in Database Systems 1985: 81-108

[Du95] W. Du, M.-C. Shan, U. Dayal, *Reducing Multidatabase Query Response Time by Tree Balancing*. SIGMOD Conference 1995: 293-303

[Evre97] C. Evrendilek, A. Dogac, S. Nural, F. Ozcan, *Multidatabase Query Optimization*. Distributed and Parallel Databases 5(1): 77-114 (1997)

[Meng95] W. Meng, and C. Yu, *Query Processing in Multidatabase Systems*, In Modern Database Systems: The Object Model, Interoperability, and Beyond, edited by W. Kim, Addison-Wesley/ACM Press, 1995:551-572

[Ozsu99] M. T. Ozsu, P. Valduriez, *Principles of Distributed Database Systems*, second edition, Chapter 15, Prentice-Hall, 1999

[Salem00] M. Mohamed Salem, G.v.Bochmann and J. Wong: "A Scalable Architecture for QoS Provision in Electronic Commerce Applications, submitted for publication.

[TPCW] TPC-W Benchmark Specification, http://www.tpc.org/wspec.html

[Urha98] T. Urhan, M. J. Franklin, L. Amsaleg: *Cost Based Query Scrambling for Initial Delays*. SIGMOD Conference 1998: 130-141

[Ye99] H. Ye, B. Kerhervé and G. v. Bochmann, Quality of service aware distributed query processing, 10th Intern. Workshop on Database & Expert Systems Applications, Florence, Italy, 1-3 Sept. 1999, Proc. published by IEEE Computer Society, 1999.

[Ye00] H. Ye, G.v. Bochmann, B. Kerherv, *An adaptive cost model for distributed query processing*, UQAM Technical Report 2000-06, May 2000

A Methodology and Implementation for Analytic Modeling in Electronic Commerce Applications

H. Keith Edwards[1], Michael A. Bauer[1], Hanan Lutfiyya[1], Yumman Chan[2], Michael Shields[2], and Peter Woo[2]

[1] University of Western Ontario, Department of Computer Science, 355 Middlesex College, London, Ontario, Canada N6A 5B8
{kedwards, bauer, hanan}@csd.uwo.ca
[2] IBM Electronic Commerce Division, 895 Don Mills Road, Toronto, Ontario, Canada M3C 1H7
{chany, mshields, peterwoo}@ca.ibm.com

Abstract. Analytic queuing theory models are used to create accurate and informative models for phenomena that range from bank teller lines to complicated distributed systems. This paper defines a methodology for creating analytic queuing theory models for electronic commerce systems. As a result of the creation of the analytic model, our research provides several important discoveries and insights into the nature of electronic commerce performance. Specifically, this paper will provide insight into the nature of the distribution of arrival times for electronic commerce systems, will indicate the organizational challenges inherent in modeling efforts, and will provide a statistical verification of web page design techniques often used by industry. Finally, this paper takes an illustrative approach and provides an implementation for an actual large-scale electronic commerce system.

1 Introduction

Electronic commerce is perhaps the major impetus behind the new economy. Industry Canada reports that the amount of global Internet commerce in 1999 was equal to \$195.39 billion CDN [8]. Furthermore, Consumer Reports Magazine reports that the amount of business to business electronic commerce is expected to increase from \$5.6 billion USD in 1998 to \$268 billion USD by 2002 [11].

With the tremendous amounts of money at stake in electronic commerce development and the lofty position allocated to electronic commerce systems in the strategic management of Fortune 500 businesses [9], there is tremendous anxiety regarding the performance of these systems. In particular, response time is a predominant though not well understood concern in these systems. The problem is that we need simple tools that enable us to both understand and improve response times in electronic commerce systems.

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 148-157, 2001.

Several recent papers focus on various methodologies and architectural strategies for improving response time in distributed systems and in electronic commerce systems [5,6,7,12,14,15]. In particular, three of these papers [5,6,12] provide illustrative analytic models as an aid in improving web application response time. While these papers provide good theoretical constructs, there are several areas that still need exploration in the realm of analytic modeling for electronic commerce that this paper will address.

In particular, no paper has yet presented a verification of the Poisson distribution of inter-arrival times for electronic commerce applications, although most papers invariably use this as an assumption. Furthermore, the approaches that do use analytic models in electronic commerce applications treat only the web server portion of the system and ignore the other components.

This paper is a preliminary investigation into the construction of an analytic model for electronic commerce. It will focus on providing a methodology for the construction of analytic models for electronic commerce systems. Such a methodology is needed to help overcome organizational complexities and to provide performance-modeling practitioners with an illustrative template for subsequent projects.

2 Challenges in the Modeling of Real World Applications

When one picks up a text on the construction of analytic queuing models[1,2], the construction of a queuing theory model would appear to be solely an exercise in constructing algebraic equations. Unfortunately, determining the model and obtaining the necessary metrics for a large-scale system is not as straightforward as these exercises indicate. In this section, we will explore three specific challenges that researchers face when constructing analytic models of real world applications.

The first challenge is to determine the model. When undertaking a model of a real world application, one does not have a characterization of the traffic within the system nor the assurance that the traffic will integrate easily into a well-defined model.

The second challenge is to find out which components reside within the system and how to isolate those components through experimentation and instrumentation. While it is tempting to treat the entire system as a black box or single entity in this situation, that approach would deprive us of many of the model's details.

The final challenge is that information can not be garnered from a single source within an organization. One must talk with several developers and subject matter experts in order to gain an understanding of the system under consideration and its inner workings. Furthermore, organizational structure and access to resources can limit the access to this information.

3 Methodology Used

In this section, we introduce the methodology used in developing our analytic model. This methodology will interest the reader who wishes to construct an analytic model

of a real world application in light of the various business challenges documented in Section 2 of this paper. We divide this methodology into four main steps with appropriate sub tasks highlighted under each major step. A stepwise methodology lends itself admirably to project management and to tracking methodologies such as Gantt charts and PERT charts.

The first step is to document and to understand the architecture of the system in order to create the most accurate model possible. In particular, information regarding the architecture of the system can be gained from three primary sources. Perhaps the best source of information on system architecture is the subject matter expert. A subject matter expert (SME) is an individual within the organization who possesses a detailed knowledge of the system and the process used to create it. A second source for information on architecture is system documentation. While garnering information from system documentation requires a bit more effort on the part of the analyst, it provides detailed information and is accessible at all times. A final source for understanding the system architecture is the system itself. This source can provide the most detailed explanation available. By combing a variety of sources and one's own intuition, it should be possible to gain an understanding of the system architecture and its components.

The second step in constructing an analytic queuing theory model is to understand the nature of the traffic that will be visiting the site. Based upon the assumptions or verifications that one makes at this stage, it is possible to begin the construction of radically different models. Common types of distributions are Poisson, hypergeometric, normal and Erlang.

In order to construct a model that most accurately mirrors the actual application, it is important to answer two questions:

1. What is the distribution of the inter-arrival times?
2. What is the rate of the arrivals?

The third step in the creation of the model is to determine the service rates for the various components of the model. The best and most accurate way to do this is to instrument each component with some sort of monitoring software in order to accurately measure its performance. An alternative to this approach is to design experiments which seek to isolate various components of the distributed system. While this is not the most trivial of tasks either, it is possible given a rudimentary understanding of the system under consideration and can provide meaningful results. When using this approach, the analyst must consider the principals of experimental design, conduct the experiments in a controlled environment, and accurately analyze the results.

The final challenge is to create the analytic model using the appropriate tool. Analytic queuing theory models are platform independent; it is possible to create a queuing theory model using the most sophisticated programming tools or to construct one with pencil and paper. The important thing in constructing the model is to utilize the results from the previous steps in the methodology. This paper creates an implementation using the Lotus 1-2-3 spreadsheet application.

4 Case Study – Employing the Methodology to Create a Real World Model

No methodology should exist in a vacuum; practitioners must be able to implement the methodology under consideration. With this in mind, this paper will now examine a utilization of the methodology to model a large-scale application in a production environment.

4.1. Target Platform Architecture – ShopIBM

In this particular case, we chose to model the ShopIBM system. ShopIBM is the portion of the IBM(R) web site that sells systems directly to consumers and businesses. In addition to selling goods and services to consumers, ShopIBM acts as a showcase for IBM's ecommerce technology. ShopIBM is a large-scale system that services approximately 300,000 requests per day. ShopIBM features thousands of products and runs on a multicultural platform that allows support for multiple countries and languages [10]. Figure 1 shows an extremely simplified architecture for the ShopIBM system. This simplification was developed in order to abstract away the system detail and to protect IBM from disclosure of confidential information.

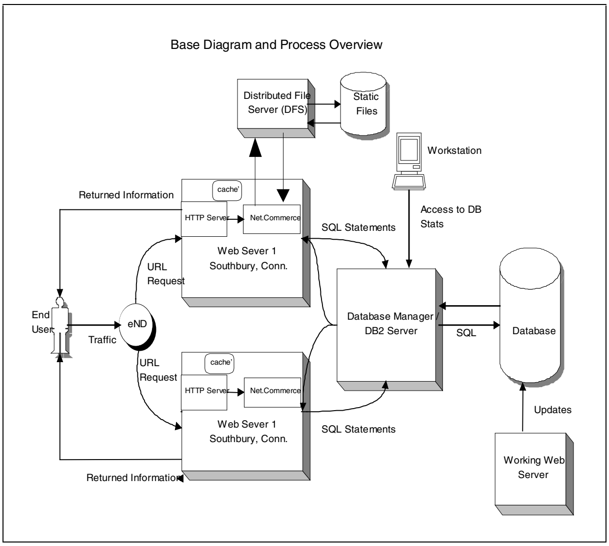

Fig. 1. The base architecture and information flow for the ShopIBM system.

The first component in this architecture is eND. The eND is the e-network distributor. It controls traffic across multiple web servers by distributing the requests in a round-robin fashion.

The second component of the basic architecture is the web server, which consists of several sub components. The most important is a program known as HTTP Server, a proprietary web server that is similar to Apache in most regards. It controls the distribution of the web pages in order to satisfy client requests. HTTP Server also generates a log that is useful for tracking the times of client requests. The HTTP Server also has a web page cache. The cache stores static web pages after the page has been requested at least one time. The reason for using non-cached pages is to avoid staleness and to take advantages of updates to the database of products.

The third component of the basic architecture is Net.Commerce. The Net.Commerce portion of the web-server is responsible for generating dynamic web pages and for performing functions such as adding an item to a shopping cart or retrieving customer information (address books, previous orders, etc.).

The fourth major component of the basic structure is the database. This is comprised of two distinct parts. The database manager transfers database entries between the web-server while the physical database stores product and customer information.

The final two components of the base architecture are the workstation that controls the database and the working web server. The workstation that controls the database provides an interface between the developers and the overall product. The working server is really a part of the development environment where changes can be safely made to the database.

Figure 2 shows the interaction diagram for the retrieval of both cached and non-cached static web pages in the Shop IBM architecture. As one can readily discern from this diagram, the retrieval of cached pages is significantly quicker than the retrieval of non-cached web pages.

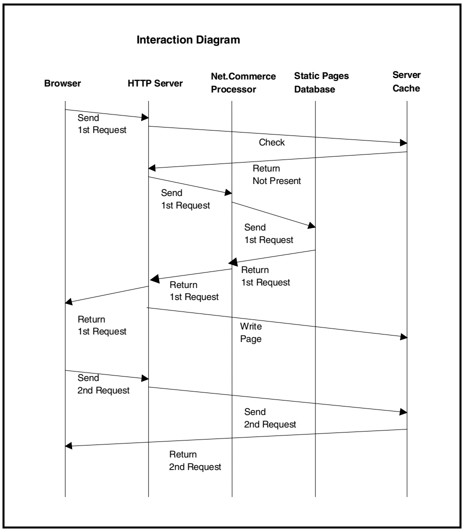

Fig. 2. Interaction diagram associated with the retrieval of static pages.

4.2 Determine Arrival Distribution and Rates through Log File Analysis

In this section, we determine the distribution of the inter-arrival rates for web server requests. We obtained log files from production servers, as traffic on the test servers might not provide an accurate reflection of the traffic patterns in electronic commerce. After obtaining the log files, we searched for each occurrence of a request for the home page and extracted the associated time stamp using AWK. While there are other ways that users can arrive at the site (e.g. deep linking), counting the home page as an arrival allows us to see the general pattern of traffic for the site.

We placed all of the arrival times for the US home page during a one day time period in a file. Next, we imported this file into a spreadsheet. The time stamps were put in chronological order and the time between them was calculated in the next column, thus obtaining the inter-arrival times. We placed the inter-arrival times into the histogram in figure 4. The inter-arrival times range from 0-60 seconds on the x-axis of the graph.

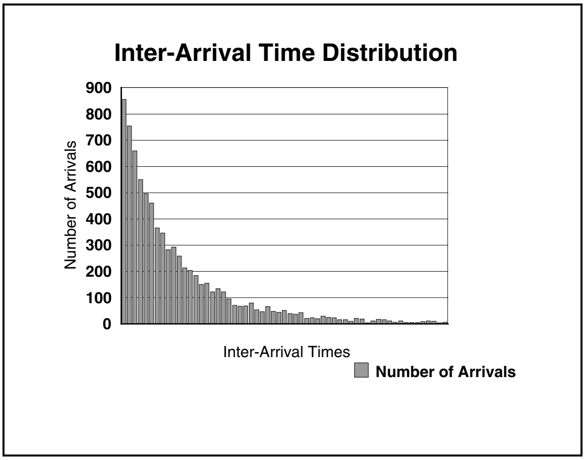

Fig. 3. Inter-arrival time distribution for arrivals at ShopIBM home page.

A run-test confirmed the independence of the observations at a 1% level of statistical significance. Since the entire population of arrival times was used for this distribution and contained no significant outliers, the distribution is identically distributed by definition. Furthermore, a Komogorov-Smirnov test showed the inter-arrival times to be consistent with an exponential distribution at both the 5% and 1% levels of statistical significance. Based upon these results, it is possible to employ an M/G/c queue for the model.

4.3 Design and Execute Experiment on Effect of Web Page Design in Relation to Server Response Time for Static Web Pages

In order to create the analytic model for static web pages, it was necessary to understand the factors that have the greatest impact on the response time for these pages. The first step is to gather anecdotal evidence and experiential insight from the subject matter experts. Based upon that dialog, the following factors were identified:

- File Size for the HTML file
- Number of Images in the document
- Whether the requested file utilized secure socket layer (SSL) protocol
- Whether the requested file resided in the server cache

These four factors were formatted into a 2^4 experimental design with three replications which will allow us to examine the variance for all factors and interactions without any confounding effects[1]. Next, we took a sample of web pages in the ShopIBM web site. These pages were categorized with respect to their file size and number of images, since it was possible to make any page reside in the cache or use SSL. The experiment was conducted in a controlled environment on a test server in order to remove factors such as network latency that could skew the results. After conducting the experiment three times, we calculated the sum of squares for each factor and conducted an analysis of variance for each factor and for the interactions of the factors. The results of the analysis of variance are given in table 1 where: factor A is the effect of caching, factor B is the effect of SSL, factor C is the effect of the number of images in the document, and factor D is the effect of the page size.

Table 1. Analysis of variance for static web page design experiment.

Factor	Sum of Squares	Variation Explained
SSA	14.1633	9.18%
SSB	30.7681	19.95%
SSC	28.3803	18.40%
SSD	26.8784	17.42%
SSAB	3.2405	2.10%
SSAC	0.9008	0.58%
SSAD	11.9838	7.77%
SSBC	2.4855	1.61%
SSBD	12.2955	7.97%
SSCD	12.2360	7.93%
3rd Order Interactions	10.3142	6.68%
Residual (SSABCD)	0.0002	0.0001%
SSE	0.6122	0.40%
SST	154.2588	100%

While the interaction of effects could not be excluded from having an effect on the over-all outcome of the experiment, the results indicated a clear disposition toward each factor having an independent effect. We also see that the amount of variance due

to the primary factors (A,B,C,D) accounted for 65% of the entire experiment. The interactions between the number of images and the other factors was approximately 7% each, accounting for 21% of the variance. This tells us that the page size, number of images, and SSL play the largest role in determining the response time for a particular page request.

Other results that were of interest from this experiment were the fact that caching the pages typically resulted in a 50% reduction in the response time. Interestingly enough, caching did not seem to have an effect for all pages, particularly when SSL protocol was being utilized.

4.4 Creation of Analytic Model

After completing the preliminary steps in the methodology, we began building the analytic model using the results of the previous steps. First, we used the architecture diagrams developed from the subject matter expert meetings to create a pictorial representation within the spreadsheet. A screen capture from the spread sheet is shown in figure 4.

The model has an intuitive user interface. The user can input the fields that are highlighted in blue and the system output will appear in the burgundy fields. This also allows the user to create a model that corresponds to their particular implementation and to validate the model based upon further experimentation. Finally, the model will dynamically update whenever an input parameter is changed. This is one of the advantages when using a spreadsheet for our implementation.

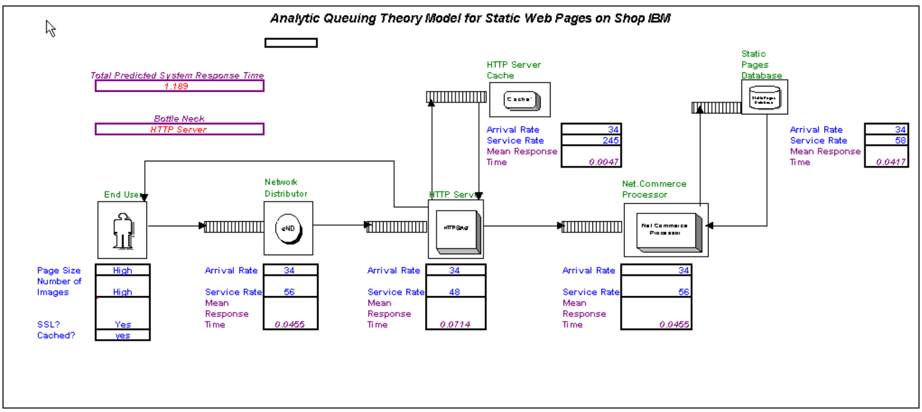

Fig. 4. Analytic Queuing Theory Model for Static Web Pages on ShopIBM.

Since we determined that arrival rates were of a negative exponential nature and possessed the properties of a Poisson distribution, we constructed our model to be an M/G/1 queue. An M/G/1 queue has several desirable properties from a modeling standpoint. Chief amongst these properties is a deterministic characterization of the mean response time, the mean number of jobs in the queue, and the utilization of a resource.

Using the spreadsheet package, we implemented the above queuing functions in a table underneath the user diagram. The end user of the model is allowed to input an arrival rate for the user requests and service rates for the various devices that comprise the distributed application. These provide the inputs for the M/G/1 Queue Properties. In addition to the arrival and service times, the user is also allowed to specify the characteristics of the requested pages based upon the experimentation done in preceding steps. In the case of static web pages, the user can indicate the following factors:

- Page Size – Low (0-60KB) or High (61+ KB)
- Number of Images – Low (0-50 Images) or High (51+ Images)
- SSL - Yes or No
- Cached - Yes or No

After the user inputs the arrival and service rates, the spreadsheet will calculate average response time based upon the rules for M/G/1 Queues, the system architecture characteristics, and the ratios of response times derived from the experiments. For example, if the type of page requested by the user takes 4.28 times as long to load as the standard page then the model will multiply the response time by that factor.

Finally, if the user requests a cached page in this static web page model, then the model will exclude the processing time due to the static pages database and the Net.Commerce commands. It will only consider the web server and the web server cache.

5 Future Work

The work done on the model thus far provides an excellent background for future research. First, we would like to develop sophisticated methods for determining individual component service rates. Secondly, we would like to express the model as a queuing network model to more accurately reflect the system. Finally, we would like to modify the model to allow the user to input a fraction of the arrivals corresponding to each category. Thus, it would be possible to predict the response time under real traffic conditions.

6 Conclusion

Analytic queuing models and precise experimentation can help us understand the performance and architecture of these distributed applications. Furthermore, these techniques can assist us in the determination of tuning parameters for the aforementioned applications. This paper provided a development methodology for analytic queuing models and showed how this methodology resulted in an analytic model of a real world application, specifically ShopIBM.

Acknowledgements. IBM, WebSphere, Net.Commerce, and ShopIBM are trademarks or registered trademarks of IBM Corp. in the U.S. and or other countries. In addition, the trademarks must be used as they are; no variations are possible.

7 References

1. Lazowksa, E., Zahorjan, J. Graham, S., Sevcik, K. Quantitative System Performance. Prentice Hall, Inc. Englewood Cliffs, New Jersey. [1984].
2. Jain, R. The Art of Computer Systems Performance Analysis. Wiley Professional Computing. Toronto, Ontario. [1991].
3. Segue, Inc. Silk Performer User's Guide. Segue Publishing. United States [1999].
4. IBM, Inc. IBM WebSphere(R) Commerce Suite Fundamentals. IBM. North York, Ontario. [2000].
5. Slothouber, L. A Model of Web Server Performance. Star Nine Technologies Incorporated. [1996].
6. Krishnamurthy, B., Willis, C. Analyzing Factors That Influence End-To-End Web Performance. AT&T Labs Research http://www9.org/w9cdrom/5/1/5/1.html.
7. Almeida, J., Almeida, V., Yates, D. Measuring the Behaviour of a World-Wide Web Server. NSF Grant CDA-9529403 and CDA-9623865. [1996].
8. Industry Canada. Canadian Internet Commerce Statistics Summary Sheet. Prepared by the Electronic Commerce Branch of Industry Canada. August 22, 2000.
9. The Means to An Edge - Ecommerce Strategic Directions. II. The Technology Vendor View. CIO Magazine. May 1, 1999.
10. Chan, Y., Suwanda, H. Designing Multinational Online Stores: Challenges, Implementation Techniques and Experience. 2nd International Electronic Commerce Conference. [2000].
11. 1998 eCommerce Report Indicates: Consumer eCommerce Segment to Increase to $26 Billion by Year 2002; Business-to-Business eCommerce to Grow to $268 Billion in Year 2002. Consumer Reports Magazine. May 20, 1998.
12. Popkov, T., Oskotski, S. Queueing Model Based QoS Management Prototype for E-commerce Systems. CASCON 2000 Proceedings. [2000].
13. Crovella, M., Bestavros, A.. Self-Similarity in World Wide Web Traffic - Evidence and Possible Causes. IEEE Transactions. [1997].
14. Manley, S., Courage, M., Seltzer, M. A Self-Scaling and Self-Configuring Benchmark for Web Servers. Network Appliance, Microsoft Corporation, Harvard. [1999].
15. Andressen, D. Yang, T. Multi-processor Scheduling with Client Resources to Improve the Response Time of WWW Applications. University of California - Santa Barbara. [1996].

Internet Based Electronic Business Framework Applications and Business to Business Standards

Deren Chen[1] and Jen-Yao Chung[2]

[1] Department of Computer Science & Engineering, Zhejiang University
P. R. China, 310027
drchen@cs.zju.edu.cn
[2] IBM T. J. Watson Research Center, Yorktown Heights
NY 10598, USA
jychung@us.ibm.com

Abstract. With the widespread popularity of the Internet, specifically the World Wide Web, Internet Electronic Commerce provides a revolutionary way of doing business, offers tremendous opportunities for business, and, most importantly, represents a market worth potentially hundreds of billions of dollars. Internet electronic commerce has become an active area recently, with many standards and solutions defined, proposed, and emerged. This paper will cover basic concepts of electronic commerce and framework of application lifecycle and three levels of approach from electronic commerce to electronic business. We will discuss recently proposed internet standards such as OTP, OBI, ICE, EDI, XML, and ebXML We will conclude with the e-marketplaces trends and directions.

1 Introduction

Internet has made the Web a valuable platform for conducting business and the application of electronic commerce is changing the traditional economic activities and human life style. Electronic Commerce (EC) is the process of linking businesses electronically with their suppliers, distributors, manufacturers, and customers to facilitate, create, or support the whole or part of the commerce cycle. Traditional commerce activities developed around seller market and concentrate the main task into promoting the sales of products. The EC is changing the traditional commerce activities and forming a new pattern of international commerce. There are four perspectives to understand EC:

- **Communications.** EC is the delivery of information, products/services, or payments via telephone lines, computer networks, or any other means.
- **Trades process.** EC is the application of technology toward the automation of trade transactions and workflow.
- **Service.** EC is a tool that addresses the desire of firms, consumers, and management to cut service costs while improving the quality of goods and increasing the speed of service delivery.
- **Online.** EC provides the capability of buying and selling products and information on the Internet and other online services.

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 158-169, 2001.

The key to EC is trade transactions over the network. Transactions are exchanges that occur when one economic entity sells a product or service to another entity. A trade transaction takes place when a product or service is transferred across technologically separable interface that links a consumer with a producer.

2 From E-Commerce to E-Business

The application of EC runs through the whole trade process from pre-trade, the trade to post-trade and forming a circle (see Figure 1). During the pre-trade, sellers need to prepare to publish/update catalog and prices, for the buyers to browse catalog and to negotiate terms. In the middle of trade, buyers place order and make payment, and the sellers will do billing and manage inventory. In the post-trade, sellers manage inventory with suppliers, coordinate with shipping company, resolve issues and maintain relationship.

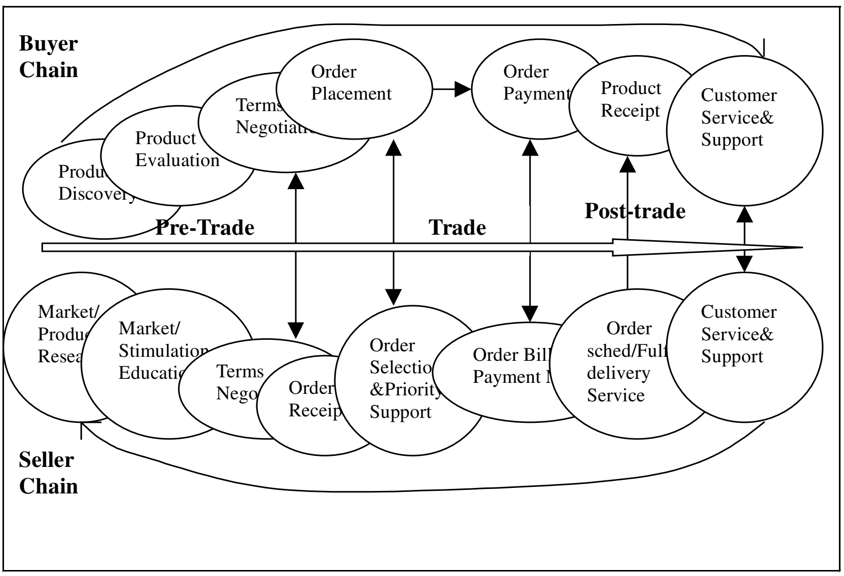

Fig. 1. Electronic Commerce Lifecycle

Internet based EC has flourished mostly in the B2C fields. There are in need of well-accepted security, technology standards and other application environment. According to the development of EC, there are three different approaches [6]:

- **Web Storefronts.** They provide a web interface to a vendor's catalog of products of services just like many software companies are advertising them as electronic business solutions. This may be an acceptable solution for B2C commerce. However customers would have to visit hundreds of suppliers' websites. This

would be an intolerable way to conduct business for a large manufacture with thousands of suppliers.

- **EC Portals.** EC Portals automate both vendor and customer buying and selling of goods and services. A major shortcoming of this approach is the security of information which resides outside of its internal firewalls while its data is being updated and maintained by a third party on the portal website.
- **E2E EC.** In this approach, every enterprise establishes its own server. All of the internal applications of different companies share information directly by some standard data interchange format such as EDI or XML (presented following). That B2B solution builds an end-to-end, enterprise-to enterprise (or E2E) EC system.

Electronic Business (EB) is wider than EC both in content and concept. Commercial activities such as buying and selling, as the term electronic commerce suggests, are certainly an important part of electronic business but they don't include the full range of EB activities (inside, outside and between) in enterprise, finance, customers, suppliers, services, government, and distribution. The following definition by Lou Gerstner, IBM's CEO, matches closely with general idea of EB: We coined the term 'e-business' to describe all the ways individuals and institutions derive value from the Net-buying and selling, but also the important transactions between teachers and students, doctors and patients, governments and citizens [9].

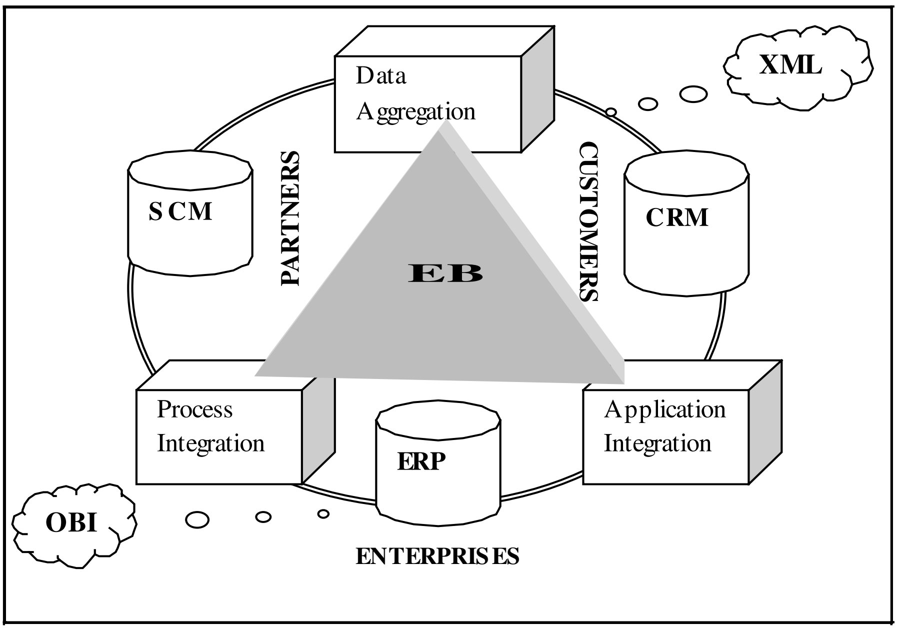

Fig. 2. EB Three Dimension Model

The goal of most EC and EB research and their associated implementations is to reduce the transaction cost in online transactions. The reduction of such transactions will enable smoother transactions between buyers, intermediaries from online

production process, and sellers. Recently, B2B marketplaces and exchanges are the emergent areas.

Customers, enterprises and partners are the three key dimensions of EB and they have their own goals: (i) maximize customer relationships for increasing loyalty and grow market share; (ii) optimize internal operations for maximizing return on investment in people and infrastruture; and (iii) maximize partner relationships for increasing collaboration via process integration. Figure 2 describes an E-Business application framework. EB is becoming critical in three interrelated dimensions.

- B2C interactions, which enables the customer to have a more direct influence in what products are made and how services (by Customer Relationship Management or CRM) as delivered intra-business interactions.
- EB enables the shift from a hierarchical command-and-control organization to the information (by Enterprise Resource Process or ERP) based organization B2B interactions.
- EB facilitates a network of loosely connected organizations where small flexible firms relying on each other to manage an integrated and/or extended supply chain (Supply Chain Management or SCM).

CRM, ERP, and SCM are each expanding their business across these traditional boundaries in E-Business environment. It's necessary for the business/commercial application to make some standards or proposals to integrate, translate, exchange information.

3 Standards and Proposal

The EB activities, such as document exchange, should be standardized across all of the different system platforms and business practices. For example, Standard General Markup Language (SGML) is a useful meta-language standard and Hyper Text Mark Up Language (HTML) which is a simple markup language and commonly is used in network environment. Electronic Data Interchange (EDI) is a common document exchange format that has been widely used among auto industry, health care providers, insurance companies, retailers, transportation, manufacturing, grocery, and financial banks. In the EB security protocol fields, there are some related standards such as Secure Electronic Transactions (SET) for secure on-line payment and Secure Socket Layer (SSL) for secure internet communication. Ongoing industrial standards mainly include Open Trading Protocol (OTP) for on-line shopping procedures; Open Buying on the Internet (OBI) for B2B non-production materials purchasing; Information and Content Exchange Protocol (ICE) for content exchange; XML/EDI for efficient EDI document exchange and Electronic Business Extensible Markup Language (ebXML) for both B2B and B2C activities.

3.1 Open Trading Protocol

The Open Trading Protocol Standard (OTP) is designed to support electronic commerce for B2C on the internet and to support different payment instrument. It

provides the means to negotiate trade, to buy and sell through underlying payment protocols, and to resolve problems. OTP provides trading protocol options which control how the trade occurs, a record of a trade, and real and virtual world delivery of goods and services. The protocol supports different types of trades including purchase, refund, value exchange, authentication, withdrawal, and deposit. The five Trading Roles, they identify the different parts which organization can take in a trade used in OTP are illustrated in Figure 3 and explained below [7].

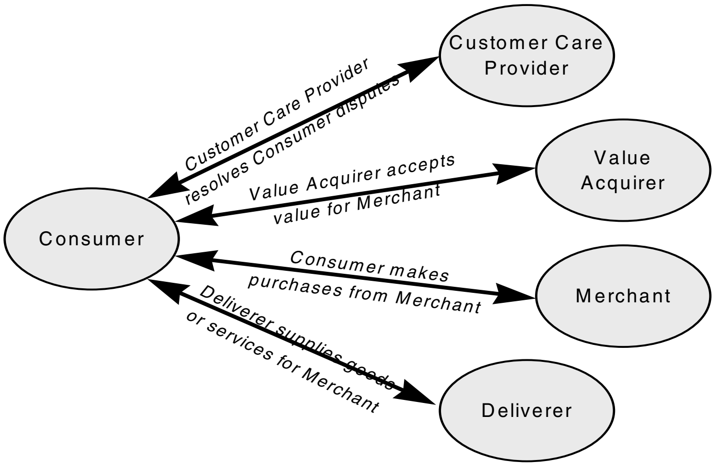

Fig. 3. OTP Trading Roles

- **The Consumer.** The person or organization that is to receive and pay for the goods or services.
- **The Merchant.** The person or organization from whom the purchase is being made and who is legally responsible for providing the goods or services and receives the benefit of the payment made.
- **The Value Acquirer.** The entity that physically receives the payment from the Consumer on behalf of the Merchant.
- **The Deliverer.** The entity that physically delivers the goods or services to the Consumer on behalf of the Merchant.
- **Customer Care Provider.** The entity that is involved with customer dispute negotiation and resolution.

These different roles may all be carried out by the merchant enterprise at the same or different internet locations. IBM, AT&T, MasterCard, and other big companies and banks take part in the development of OTP. Version 0.1 of the OTP is an initial draft for comment and was issued by Open Trading Protocol Consortium in February 1997. OTP V0.9 (Draft for Public Comment) was issued in January 1998 which including Business Description Part and Specification Part. The latest version is OTP V0.99 published in 1999.

3.2 Open Buying on Internet

In October 1996, several Fortune 500 buying and selling organizations formed a roundtable to create an open, vendor-neutral, scalable and secure interoperable standard for B2B electronic commerce. The result of that roundtable was the formation of the Open Buying on the Internet (OBI) Consortium. There are more than 80 enterprises of OBI members including IBM, Oracle, Microsoft, and Netscape. OBI standard is an open, flexible framework for B2B internet commerce solution. The initial focus of OBI is on automating the high-volume, low-dollar transactions between trading partners that account for 80% of most organization' purchasing activities. Version 1.0 of the OBI standard was published in May 1997 and included the OBI business vision, business requirements, architecture, and technical specifications. OBI V1.1 and OBI V2.0 were issued in June 1998 and August 1999, respectively. The latest version is OBI XML V3.0 released in 2000[3].

The purpose of the OBI is to provide a standard framework for secure and interoperable B2B internet commerce with an initial focus on automating high-volume, low-dollar transactions between trading partners. There are four essentials entities involved in an OBI system[2]. The **Buying Organization** procures items as part of its daily business operations. The **Requisitioner**, a member of the buying organization, is interested in procuring certain items as part of the non-mission critical process of the organization within his/her command. The **Selling Organization** supplies goods and services to other businesses. The **Payment Authority**, which may not exist in an OBI scenarios, as a clearing-house for all payment and settlement activities between the selling and buying organizations. All the aforementioned entities should be connecting to the Internet and have digital certificates that uniquely and securely establish their identities. The whole OBI architecture is based on the following model of B2B commerce:

1. A requisitioner, using a Web browser, connects to a local purchasing server located at the Buying Organization and selects a hyperlink to a Selling Organization's merchant server containing an on-line catalog of goods and services.
2. The Selling Organization's server authenticates the requisitioner's identity and organizational affiliation based on information presented in the requisitioner's digital certificate. Authentication information is used, in conjunction with profile information optionally presented by the requisitioner's browser, to uniquely identify the requisitioner and to construct a specialized catalog view. The requisitioner browses the catalog, select items, and "checks out."
3. The content of the requisitioner's "shopping basket" and identity of the requisitioner is mapped into an order request (EDI-compatible). A digital signature is calculated (optionally); the order request (and digital signature if used) is encapsulated in an *OBI* object that is encoded and transmitted securely to the Buying Organization over the Internet using HTTP and SSL. There are two alternative methods for transmitting an encoded OBI object containing an order request over the Internet using HTTP. These are referred to as the server-to-server method (step 3 in Figure 4) and the server-browser-server method (step 3a-3b in Figure 4). The Buying Organization server receives the encoded OBI object, decodes it, extracts the order request, verifies the signature (if appropriate) and translates the order request into an internal format for processing.

4. Administrative information (including payment type) is added to the order request at the Buying Organization (automatically from a profile database and/or manually by the Requisitioner), and the order is processed internally either automatically or through a workflow-based process.
5. The completed and approved order is formatted as an *OBI order* (EDI-compatible) and a digital signature is calculated if desired. The order (and digital signature if appropriate) is encapsulated in an OBI object that is encoded for transport and transmitted securely from Buying Organization server to Selling Organization server via the Internet using HTTP over SSL. The Selling Organization receives the encoded OBI object, decodes it, extracts the order, verifies the signature (if appropriate), and translates the order into its internal format.
6. The Selling Organization obtains credit authorization, if necessary, and begins order fulfillment.
7. The payment authority issues an invoice and receives payment.

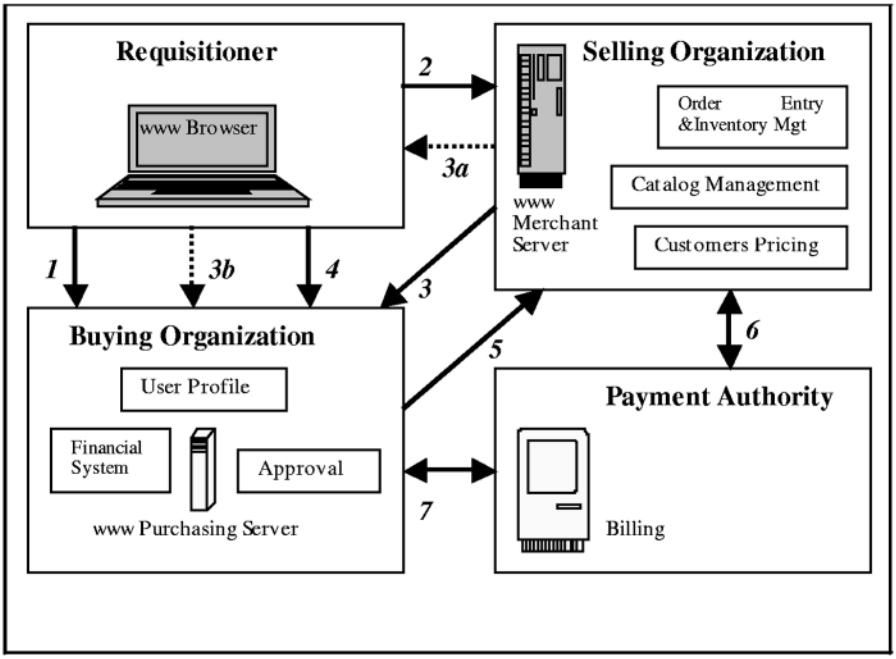

Fig. 4. OBI Architecture

3.3 Information and Content Exchange

The Information and Content Exchange (ICE) Protocol is a B2B protocol for exchanging document between content provider and content distributor. The goal of ICE is to reduce the cost of doing business online and increase the value of B2B

relationships[8]. ICE facilitates the controlled exchange and management of electronic assets between networked partners and affiliates.

ICE covers four general types of operations: Subscription Establishment and Management, Data Delivery, Event Logs, and Miscellaneous. There are two entities involved in forming a business relationship where ICE is used: a **Syndicator** and a **Subscriber**. A relationship between a syndicator and a subscriber starts off with some form of subscription establishment. In ICE, the subscriber typically begins by obtaining a catalog of possible subscriptions offers from the syndicator. The subscriber then subscribes to particular subscriptions, possibly engaging in protocol parameter negotiation to arrive at mutually agreeable delivery methods and schedules. The relationship then moves on to the steady state, where the primary message exchanges center on data delivery. ICE uses a package concept as a container mechanism for generic data items. It defines a sequenced package model allowing syndicators to support both incremental and full update models. ICE also defines push and pull data transfer models for server-to-server protocol. The latest ICE is V1.1 (Review R) issued in May 2000 by ICE Authoring Group including Vignette Corporation and other enterprises.

3.4 XML/EDI

Electronic Data Interchange (EDI) is a set of specification for formatting machine-readable documents that is designed to automate business flow among businesses by replacing paper documents (such as purchase orders and invoices) with paperless ones. Traditional EDI system contains two major components: EDI translation software that converts and maps EDI formats to/from internal business applications, and communication channels that deliver EDI documents to the desired trading partners[1]. Over these years, some different industries and countries have developed their own EDI standards for representing ANSI X12 (US standard) or EDIFACT (international one). Although EDI also has been successfully employed in specific industries (such as retail) and in some large enterprises, it has not been widely adopted. The primary barriers to widespread acceptance of EDI are the costs of implementation and the costs of communication, which is frequently implemented by using Value Added Networks (VANs). These costs are generally too high for companies that do not conduct large numbers of EDI transactions.

The eXtensible Markup Language (XML) is an initiative proposed by the W3C as an alternative to HTML which currently dominates web publishing. Unlike HTML, XML is a meta-language - a language that allows you to create your own markup languages for your purposes. There are three essential factors in XML: Document Type Definition (DTD or XML Schema), eXtensible Stylesheet Language (XSL) and eXtensible Link Language (Xlink). DTD and XML Schema define the logic structure of XML files, elements or their attributes and the relation between them. XSL defines the grammar normalized while Xlink describes the link between each resources based web. The application of XML involves producing and parsing (by parser using DOM or SAX standards).

The difference between EDI and XML is summarized in Table 1.

Table 1. XML and EDI e-Commerce solutions compared

EC solution	XML	EDI
Optimized for	Easy programming	For compressed messages
Server costing	Requires standard web server	Dedicated EDI server, more costly
connection	Uses your existing Internet	Uses VAN and charged by usage
Message format	Easy to learn, tool available	Takes time to learn and manipulate
Language	Only requires JavaScript, VB, Python or Perl script writers	Requires trained programmers
Programming	Easy to read and debug	Difficult to read and debug

Figure 5 provides a sample XML purchase order. It's both machine and human readable while the EDI document presenting same purchase order is only machine-readable [6].

There are many useful features for the application and development of XML such as extensible, cheaper for related software, chance to unify and simplify, more strongly typed data representation, utilize well-defined DOM functions, well interface for EDI, more convenient for development web-based. All of these standards mentioned such as OTP, ICE and OBI (V3.0) have supported or will support XML format.

The XML/EDI initiative is to provide a common framework, based on XML, for those messages formats EDI provided and to leverage existing tools for the production and processing of that information. The combination of XML with EDI holds the promise of extending the advantages of Web-based EDI through an open standard to the millions of small and medium sized enterprises.

3.5 Electronic Business XML (ebXML)

Electronic Business Extensible Markup Language (ebXML) is an international initiative established by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS). The purpose of the ebXML is to provide an XML-based open technical framework to enable XML to be utilized in a consistent and uniform manner for the exchange of EB data in application to application, application to human, and human to application environments—thus creating a single global market [4].

```
<?xml version="1.0" ?>
  <?xml:stylesheet?>
  <purchase-order
  <header>
  <po-number>1234</po-number>
  <date>1999-02-08</date><time>14:05</time>
  </header>
  <billing>
  <company>XYZ Supply Store</company>
  <address>
  <street>601 Pennsylvania Ave. NW</street>
  <street>Suite 900</street>
  <city>Washington</city><st>DC</st>
     <postcode>20004</postcode>
  </address>
  </billing>
  <order items="1" >
  <item>
  <reference>097251</reference>
  <description>Widgets</description>
  <quantity>4</quantity>
  <unit-price>11.99</unit-price>
                <price>47.96</price>
  </item>
  <tax type="sales" >
  <tax-unit>VA</tax-unit>
  <calculation>0.045</calculation>
  <amount>2.16</amount>
  </tax>
                           ...
```

Fig. 5. Sample XML purchase order

ebXML is based on international standards and is itself intended to become an international standard. The scope of the ebXML business requirements is to meet the needs of both B2B and B2C activities. General ebXML principles to be followed in developing ebXML deliverables are to create technical specifications that:

1. Enable simple, easy and ubiquitous electronic business through use of XML;
2. Use XML technical specifications to the maximum extent practicable;
3. Provide a global cross-industry open/interoperable standard for B2B and B2C trade;
4. Coalesce the structure and content components of divergent XML initiatives into a single useable XML business standard;
5. Provide impetus so that common resources currently engaged in short-term; solutions shall be marshaled to reach a common long-term solution goal;
6. Support vertical and horizontal segments of industry and business participants;

7. Avoid proprietary solutions that impose financial or software requirements constraints on ebXML users to buy, install or programmatically support any ebXML unique software products in the conduct of business information exchange;
8. Strive to minimize costs of doing business electronically;
9. Provide multi-lingual support;
10. Accommodate national and international trade requirements;
11. Provide a migration path from accredited EDI and developing XML business standards;
12. Apply when possible the simplification principles of SIMAC Business Requirements.

4 E-Business Solution Based Standardized Products

Several important issues which are critical for wide spread adoption of EB includes merchant-side solutions and procurement-side solutions based on related standards in trading protocol (such as OBI), interoperability and exchange (such as XML/EDI), security and confidentiality (such as SET) with these solutions. Most sell-side applications involve component solutions, such as catalog, language translation, supply chain management, payment processes, shipping and handling, customer service and support, customer profiling analysis, Web site analysis, and security. Total solution applications can provide a single set of solutions for companies to sell their products over the Internet. Buy-side applications can enable companies to purchase products and services over the Internet from a web-enabled supplier. Procurement solutions should include workflow approval processes, integration to back-end enterprise resource planning systems and accounting systems, user authentication and identification, and secure payment protocols. A successful E-Business application solution needs an open, standards-based platform, design patterns for E-Business, and a comprehensive set of leadership products.

5 Future Trends and Summary

EB is moving from experimental prototype to business-critical system. The growth of E-Business expectations and requirements will be exponential, revenue will be doubling every 12-18 months, especially in B2B marketplaces and exchanges. EB is not only focus on these primary fields such as business, economics, communication, computer (software and hardware) engineering and applications, security, transactions, but also more and more related disciplines including law, politics, anthropology, sociology, organization theory, process engineering, linguistics, and collaboration multimedia. EB progresses driven by combination of these fields and standardized based solution will be the future of EB.

There are several evolutionary trends of E-Business such as: (i) from buy-side and sell-side centric to B2B marketplace and exchange, (ii) from fixed price model to the dynamic pricing including auctions, reverse auction and dynamically negotiated pricing, (iii) from standard catalog buying to build to order and design to order, (iv)

from one to one interaction to many to many interactions, (v) from the pre-planed model to the on-line decision support and deep computing, (vi) from the static HTML to dynamic HTML then to the XML based JSP, and (vii) the integrated supply chain management, collaborative and mobile commerce.

References

1. S. Fu, J.-Y. Chung, W. Dietrich, V. Gottemukkala, M. Cohen, and S. Chenetc: A Practical Approach to Web-based Internet EDI, Proceedings of 19^{th} IEEE International Conference on Distributed Computing System (ICDCS) Workshops on Electronic Commerce, pp.53-58, May 1999.
2. Z. Tian, J.-Y. Chung, L. Y. Liu, V. Guttemukkala, J. Li: Business-to-Business e-Commerce with Open Buying on the Internet, Journal of Netnomics, Fall 2000.
3. The OBI Consortium: "Open Buying on the Internet (OBI) Technical Specifications, Release V2.1, November 1999, http://www.obi.org
4. The ebXML Technical Requirements Project Team: ebXML Requirements Specification Version 1.0, May 2000, http://www.ebxml.org
5. Robert S. Sutor: IBM XML Industry Standards Liaison, IBM Japan XML Day, August 1999.
6. XMLSolutions Corporation: XML and EDI: Peaceful Co-Existence, 1999, http://www.xml.org
7. The Open Trading Protocol Consortium: Internet Open Trading Protocol Version 0.9, January 1998, http://www.otp.org
8. ICE Authoring Group: The Information and Content Exchange (ICE) Protocol AC Review Version 1.1 (May 2000), Revision R, http://www.icestandard.org.
9. Berthold Daum, Markus Scheller: Success with Electronic Business, Addison-Wesley, Great Britain, 2000.

Online Auction Protocols: A Comparative Study

Carsten Passch[1], William Song[*,2], Weidong Kou[2], and Chung-Jen Tan[2]

[1]Information Technique Department, Dresdner Bank, Hong Kong
Carsten.Paasch@dresdner-bank.com
[2]E-Business Technology Institute, The University of Hong Kong
{wsong,wdkou,ctan}@eti.hku.hk
Hong Kong, China

Abstract. Online auction is an interesting component of the electronic commerce. Goods and services that used to be sold in conventional brick-and-mortar stores at fixed prices can now be purchased online at their "true" valuations in the eyes of multiple customers. With the growing sheer number and total value of these transactions conducted under the online auction model, the actual security becomes a controversial issue. This is where cryptographic methods, some already decades old, others brand-new and specialized for the application in the realm of auctions, come into the game. This paper attempts to give an overview of the current state of affairs and convergent issues for cryptographic methods applicable to the field of on-line auctions.

1 Introduction

Auctions on the Internet have become a fascinating new type of exchange mechanism. Everyday, hundreds of thousands of different auctions take place online, for goods ranging from Nike shoes and Gucci handbags to used parachutes. Internet technology has lowered the costs of organizing auctions and of participation as a bidder in them, which seems to be the cause for auctions being used for more and more transactions over time. Indeed, online auctions currently trade billions of US dollars' worth of goods per year and are growing at a rate of more than 10% per month [5]. These facts demonstrate that online auctions are a significant part of the evolving e-commerce universe that is worth studying with diligence from a variety of perspectives.

In the following, we briefly discuss online auction problems, which are related to cryptographic methods, the properties of online auction transactions, and how cryptographic methods support online auctions.

1.1 Online Auction Problems Related to Cryptography

Online fraud [7] is said to have increased from 1997 to 1998 by 600 per cent. The number one culprit in this area was online auctions. According to [18], fraud tends to happen in low-dollar amounts, but due to their large quantity the total amount of the

---

[*] The corresponding address is William Song, E-Business Technology Institute (ETI), The University of Hong Kong, Pok Fu Lam Road, Hong Kong. Email: wsong@eti.hku.hk.

W. Kou, Y. Yesha, and C.J. Tan (Eds.): ISEC 2001, LNCS 2040, pp. 170-186, 2001.

frauds quickly rises. Contradicting her point, Lucking-Realey [5] claims that there is very little fraud in online actions following the number of public reports about such cases. A possible explanation for this contradiction can be that there may be a number of small fraud cases with low dollar amounts at stake that never make it to the headlines. Following the argument, however, that online auctions will become one of the major applications on the Internet, it is necessary to understand security requirements in online auctions as the absolute number of fraud cases is likely to increase over time.

The survey conducted on online auctions in late 1998 has revealed that security and encryption were not a focus of much attention at that time [5]. Even companies, which accept credit card numbers as forms of bid payments submitted through online forms and sent over the Internet, do not always secure the transactions. The survey found that only 27% of the sites surveyed showed some form of encryption or security, and over half of the sites made no mention of the security issue at all.

The reason why the security and privacy issues have not been treated seriously for the moment can be that the majority of online auction sites still serve the Consumer to Consumer (C2C) or Business to Consumer (B2B) market where transaction sizes are generally small and the entertainment aspect of the online auctions is dominant. As we argued earlier, this state of affairs will certainly change, however, when business to business (B2B) electronic commerce becomes the dominant driving force in the online markets where transaction sizes are generally higher and competition between market participants demands a more stringent approach to security aspects. This effect will also have an impact on the structure of online auctions.

1.2 Properties for Online Auction Transactions

Several properties have been identified as either required or desirable in protocols for transactions in electronic commence [9] and thereby implicitly for online auctions. These properties are:

- Security: Passive or active attacks by an intruder, abnormal termination of the protocol by involved parties, or corruption or loss of messages in the network should not cost more than some additional computation to either party
- Privacy: No third party should be able to know details about the transaction (payment amount, product details, identity of parties that wish to remain anonymous etc.) unless it has been part of the product advertisement available to all
- Anonymity: Nobody (including the merchant involved in the transaction) should know the true identity of the customer involved. At least, nobody should be able to trace all transactions made by a customer and create a profile of the customer. This is desirable in many forms of electronic commerce, e.g. in investment into securities (see also [6]), and also in Internet auction. It is desirable for auction participants to maintain their anonymity until the auction is closed. However, it is shown in [17] that if bidders can submit bids under false names or multiple bids un-

der fictitious names, there exists no auction protocol that simultaneously satisfies incentive compatibility,

- Atomicity: Under all circumstance, the transaction must either go to completion (the right goods are exchanged for the right amount of money) or be aborted (there is no exchange of goods or money). A good discussion about the specifics of atomicity in electronic commerce can be found in [12].
- Low Overhead: The cost for each transaction must be low enough to justify transactions involving small amounts of money.

The auction house policy and the instructions from the seller dictate whether the auction is accessible to the public at large, to the buyers and sellers registered with the auction service, or only to buyers registered to participate in the current auction. Access control mechanisms are needed to enforce these rules. Security mechanisms are needed to ensure that an outsider does not sabotage the site announcing the auction and the auction rules. This includes preventing unauthorized postings and alterations as well as preventing denial of service attacks. A possibility for achieving this is to use a trusted third party.

1.3 Cryptographic Support

Cryptographic mechanisms, used to verify a particular auction notice being posted and accessible during a certain time period will be very useful in auctions processes. During the bidding phase cryptographic mechanisms are needed to ensure that a bid submitted is not tempered with, or disclosed to other bidders in violation of the auction rules. In open outcry auctions, spurious bids, injected by the seller or auctioneer to prompt the highest bidder to further increase his bids, must be prevented by establishing a verifiable connection from every bid to a known bidder. In the real world such unethical behavior is called "taking bids off the wall or ceiling". A shill is a human agent deployed to inject spurious bids into an auction.

Spurious or phantom bids are possible in real auctions because knowledgeable or well-known buyers often want to remain anonymous. The presence of knowledgeable bidders in an auction prompts other bidders to bid high undermining the interest of the knowledgeable bidders. Accommodation of knowledgeable bidders gives the auctioneer an excuse to pick bids from "nowhere". Internet auctions cannot overcome this mechanism design constraint. The problem is further aggravated because Internet auctions will have much larger participation from geographically dispersed bidders and cyber identities can be created easily. The possible approaches to this problem are either caveat emptor, requiring mechanisms to let bidders establish the identity of other bidders, or an independent third party trust rating system that can investigate the ethical behavior of the auctioneer and assign a trustworthiness rating to it. Shills are detectable by experienced bidders because when shills are the winner in an auction, which happens sometimes accidentally, the items reappear on the auction block. Shills appear more frequently at auctions and tend top bid on unrelated items. However, it is easy to hide these clues on the Internet.

1.4 Overview

Although there are many auction protocols available, there lacks a systematic comparison between them, in particular, in terms of their characteristics. In the paper, we extract some criteria from the auction protocols and use these criteria to make comparison study on the auction protocols. The rest of paper is organized as follows. In the next section, we describe the current auction protocols. Then in section 3, we propose a collection of criteria, which are used as a basis for the protocol comparison. We conclude the paper in section 4.

2 Online Auction Protocols

In this section, we shall explain those auction protocols that appear and are cited most frequently in the academic literature. For each protocol we shall briefly give an outline of its purpose and its basic functionality. The underlying fundamental cryptographic methods are described as appropriate together with the protocol. For further references of underlying technologies, the reader is advised to consult the papers in which the protocol has been published originally.

Different auction protocols have been designed to reach different design goals and requirements in the real life. We will introduce the most frequently cited auction protocols in the relevant research literature, grouping them into families of protocols that have similar design goals in common, e.g. protocols that address the requirements of secret bid auctions, English auctions, or general-purpose protocols. Other protocols are mostly geared at addressing timing issues of Internet auctions. We shall give a sketch of the functionality of all these protocols and draw the readers' attention to the special features that each of these protocols has. It should be noted that the protocols introduced make use of well-known cryptographic mechanisms such as RSA, SSL, DES etc. The protocols that we introduce assume these standard primitives are well known, and we will therefore not explain them in further detail. Please also note that due to the complexity of some of the protocols and since we try to show as many protocols that have appeared in the literature as is possible, we can sometimes only sketch their functionality, without going into great detail.

2.1 Auction Protocols for Sealed-Bid Auctions

The protocols introduced here are specialized to the implementation of sealed bid auctions in a secure fashion. A sealed bid auction is one in which secret bids are issues for an advertised item, and once the bidding period closes, the bids are opened and the winner is determined according to some publicly known rule (e.g. the highest bidder wins). Sealed-bid auctions are used for example in the auctioning of mineral rights on U.S. government-owned land, or for the sale of real estate. They pose some novel security problems [2]:

1. Central to the fairness of a sealed bid auction is the secrecy of sealed bids prior to the close of the bidding period. This means that the timing of the disclosure of bids is crucial.
2. Auctions require non-repudiation mechanisms to ensure that payments can be collected from winning bidders.
3. Due to the secrecy surrounding sealed bid auctions, it may difficult for outsiders to have confidence in the validity of the auction.
4. Some types of sealed bid auctions should enable bidders to remain anonymous.

These problems are only exacerbated when one considers the implementation of auctions on distributed computer systems, or the possibility of a corrupt insider in the auction house collaborating with bidders.

It is worth noting that some of the problems described above could arise in a sealed bid auction simply due to the benign failure of the auction service or the bidding process, e.g. a non-fault-tolerant auction service could be collecting money from the winning bidder without granting the auctioned item.

Informally, the cryptographic interesting part of a sealed bid auction consists of two phases of execution. The first phase is the bidding period, during which arbitrarily many bidders can submit arbitrarily many sealed bids to the auction. At some point, the bidding period is closed, thus initiating the second phase in which the bids are opened and the winner is determined and possibly announced, presumably this is the highest bidder.

The auction service should provide properties that can be subsumed under the two categories of validity and secrecy [2].

Validity

- The bidding eventually closes.
- There is at most one winning bid per auction, dictated by the (deterministic) publicly known rule applied to well-formed bids before the end of the bidding period.
- The auction service collects payment from the winner equal to the amount of the winning bid.
- Correct losing bidders forfeit no money
- Only the winning bidder can collect the item bid.

Secrecy

- The identity of a correct bidder and the amount of its bid are not revealed to any party until after the bidding period is closed.
- The protocol can also be modified to allow for submission of anonymous bids.

This discussion of the properties does not address attacks that involve collaboration among bidders to “fix” the price that wins the auction, i.e. auction rings. However, the possibility of having rings in sealed bid auctions is lower in any case, since cheaters to the ring would remain anonymous. It also does not address attacks in which messages to and from bidders are intercepted, delayed, or otherwise manipulated in transit. These attacks, however, have no effect on the validity or secrecy properties described above.

The following five auction protocols belong to the auction category of sealed-bid auctions.

Protocol 1: The Millionaires' Protocol. The protocol introduced here is of rather simplistic nature. It is based on Millionaire's protocol that allows two rich men to compare their wealth to determine who is richer, without actually revealing the amount of their assets to each other [15]. It has been described both in [8] and [13], and can probably be considered one of the classics in the realm of online auction protocols. It works as follows. An auctioneer A receives bids from n bidders $B_1 \ldots B_n$. Their encryption and decryption functions are denoted as e_A, $e_1 \ldots e_n$ and d_A, $d_1 \ldots . d_n$. A informs each bidder B_i of the following bidder B_{i+1}. The possible bids are represented by integers in some interval. With n bidders, we need n-1 repetitions to determine which bid is the highest. Once the highest bid has been found, the corresponding bidder identifies himself to the bid-taker who then asks him to reveal his private encryption function. Thereafter, the price of the object can be determined.

Protocol 2: Sealed Bid Auction based on Verifiable Signature Sharing. In [2], the authors propose a secure distributed auction service that supports the submission of monetary bids for an auction and ensures the validity of the outcome, despite the malicious collaboration of arbitrarily many bidders and fewer than one-third of the auction servers comprising the service. The auction service is guaranteed to declare the proper winning bidder, and to collect payment in the form of digital cash from only that bidder. It is guaranteed that no bid is revealed prior to the close of the bidding period. It is also possible for bidders to submit anonymous bids. The resilience of the service to malicious auction servers can be leveraged to provide resilience to malfeasant auction house insiders. If each individual is allowed to access less than one-third of the servers (e.g. through spatial or administrative separation), then corrupting an insider provides no advantage to the bidder of the auction. This reduces the incentive for "buying off" insiders in the auction house. The approach employs a range of old and new cryptographic techniques. The resulting system demonstrates novel and efficient methods for protecting electronic currency in competitive environments. It also provides insights into addressing similar issues in other types of auctions.

Protocol 3: Sealed Bid Auctions with Secret-Sharing Shared Polynomials. In [3], the authors describe an auction protocol for sealed bid second-price (Vickrey) auctions that seeks to address the design criteria of being economical, executing fast and, above all, maintaining privacy. The economic design implies that the auction should be conducted based on sound economic principles, avoiding the possibility that the final winning bid may be artificially low. With some additional modifications, also the design goal of maintaining bidders' anonymity can be achieved.

Protocol 4: Multi Round Sealed Bid Auctions. The protocol described in [2] above unfortunately results in all auctioneers knowing all bids after the auction is decided. In [3] the authors propose a protocol based on secure distributed computation, which hides all bids except for the winning bid. The millionaires protocol [15] used earlier is an example of how to use such a secure distributed computation.

The suggested protocol uses a distributed set of m auctioneers, so that any m-1 of them cannot open a bid, similar to the protocol in [2]. The protocol only deals with passive attacks, i.e. groups of auctioneers or eavesdroppers who collaborate on information. Active attacks, i.e. auctioneers who might attempt to actively lie about the values they receive, are not addressed. However, the values of specific bids are kept secret even at the termination of the auction. Each round of the auction has a constant

number of communication phases. In each round of the auction, bidders can place a bid for a constant number of values k. For example, when bidding for an item, the first round of the auction may have k=10 auction values of $100, $200, …, $1000. If the first round of the auction results in the maximum being a tie for a value of, say, $400, then bids are placed for a refined auction of $400, $410, …, $490. As k is increased, the size of each bid increases, but as k is decreased, the likelihood of multiple rounds is increased. The trick is to find optimal values for k. The protocol goes beyond secret auctions to implement extremely secret auctions. The result is a hybrid of a sealed-bid auction and an English auction, since possibly more than one round of bidding can occur. The protocol seeks to ensure non-repudiation, privacy and efficiency.

Protocol 5: Homomorphic Encryption with Φ-Hiding Assumption. This protocol is described in [1]. It starts out with a general problem description: A wants to buy some good from B if the price is less than . B would like to sell, but only if the price is more than , and neither one of them wants to reveal these secret bounds. To achieve a resolution for this, the protocol uses an oblivious third party T who learns no information about or , not even whether > . The protocol requires only a single round of interaction and ensures fairness. It essentially addresses the problem of private bidding.

2.2 Protocols for English Auctions and Reverse Auctions

Assumedly, due to the higher secrecy and privacy requirements for sealed-bid auctions, the number of cryptographic protocols that have been developed for them is larger than for any other auction type, even for English auctions. In a way this is surprising, though, given that English type auctions are by far the most popular ones on the Internet. A few papers, however, have been published that are dealing with English-type auction and we will introduce two auction protocols here.

Protocol 6: No Special Trusted Parties using Reverse Hash Chains. An English auction protocol is presented in [11]. The interesting feature here is that this protocol works without special trusted third parties. There is also no need to trust the auctioneer to obtain a fair outcome of the auction. This is normally a problem in existing auctions. Without detection, the auctioneer may selectively block bids based on the bidder and amount. Also, the bidders must trust that the auctioneer does not selectively close the auction after a particular bid is received. The auction methods stated above depend on trusted auctioneers, or at least on a number of trusted auction servers, which can be difficult to enforce, especially in smaller auction houses. In [4] it is stated in the context that "(…) in simplistic terms, it is reasonable to expect that, say, three out of five employees (servers or server administrators) in a government organization or xyz_megacorp will be honest. But it is different to assume that three out of five servers deployed by the relatively little xyz_little_corp will not collude." It would therefore be useful to also have assurance in the necessary trust for auctions conducted by small auction houses – or better still to remove the trust entirely.

Protocol 7: An English Auction-like Reverse Auction Protocol. Agent based trading is one of the applications that is facilitated greatly by Internet auctions. Instead of

human auction participants, electronic software agents perform this role instead. Here we will show an example of a protocol of what the authors call an "agent-based English auction-like negotiation protocol" that has been developed to work with these agents. It not only retains the agent's mobility and flexibility, but also takes secure measures to prevent attacks from malicious hosts during the negotiation process. The protocol is presented in its generic form in [16] and further developed for application in Internet retail commerce in [14].

2.3 General Purpose Protocols

After the above description of various auction protocols that focus on particular issues that arise in online auctions (e.g. some focusing on server collusion, others looking more at efficiency of implementations) and generally being useful for only a particular type of auction, we will introduce in this subsection a protocol that is a general purpose protocol for online auctions. The extensive work in the area of online auctions can be found in [9].

Protocol 8: A Secure General-Purpose Auction Protocol. This protocol has been developed only in 1998 and is described embedded in the wider context of secure E-commerce protocols. The protocol is described in detail in [9]. This protocol indeed is the only general-purpose online auction protocol that we could find in the literature today at best research efforts. The interesting aspect of this protocol is the fact that it addresses the requirements that have been stated earlier as being desirable for Internet auctions: security, privacy, anonymity, atomicity and low overhead, i.e. efficiency. The protocol is applicable for fully automated electronic auctions between anonymous customers and a merchant whose identity is public. It will be shown that also in the presence of a powerful attacker capable of passive and active attacks the protocol fulfils the requirements.

3 Comparative Study of the Online Auction Protocols

We have described and analyzed some typical online auction protocols in the previous section, and shall now try to draw a comparison between these protocols in order to gain further and deeper understanding of the current status of the online auction protocols. In the following, we shall compare the protocols along the ten dimensions of:

- Supported auction types
- Level of complexity
- Security
- Anonymity
- Privacy
- Atomicity
- Efficiency
- Incorporation of timing issues
- Ease of implementation
- Addressing issue of server collusion

We will refer to the auction protocols that have been presented earlier by their number (i.e. P1-P8 for protocol one through eight). We will generally follow the format of going through all 8 protocols and describing how each particular feature is addressed in that protocol.

3.1 Supported Auction Types

Expectedly, the number of auction types that are supported varies; some of the protocols can be considered more like generic building blocks that can be deployed as part of a wider auction mechanism, for example P1, others are ready-made protocols for a specific type of auction, such as P2. We will go through all eight introduced protocols and show what they can be used for.

P1 is essentially a protocol that enables two people to compare two numbers without revealing these numbers to each other. The core of the idea is keeping the bids secret from other bidders. In that sense P1 is mostly applicable to secret-bid auctions. The protocol can be easily modified to handle also Vickrey auctions. It is also very simple to set reserve prices. The protocol may be tweaked around to be able to support English auctions as well.

P2 is a fully functioning protocol that is specifically designed to implement secret bid auctions. Generally, a secret bid auction is the type that most auction protocols have been designed for. P2 does not support any other type of auction.

P3 has been designed for a sealed-bid auction and can support either first price or second price (Vickrey) type. The introduction of the presentation of the protocol is done using the Vickrey type. The protocol does not support any other type of auction.

The protocol P4 is originally designed for single round sealed bid auctions (first or second price), however, it can also be used to simulate a number of other auctions, which are essentially a hybrid of a sealed bid auction and other types, e.g. secret English auction, secret Dutch auction, or even binary tree auction or hierarchical auction.

P5 is in fact a replacement algorithm for P1; it also determines the higher of two numbers without revealing them to their owners. It is therefore most appropriate for secret bidding. The authors in fact give an idea on how to implement a sealed-bid auction based on this protocol. This can be first or second price, since the protocol yields simply a partial order of the received bids as the computation result, so the first or second highest bid can be identified.

P6 is a protocol that implements auctions of the English type. No other auction formats are supported by this protocol.

P7 is a bit of an odd-man-out in this comparison since it is the only protocol that is specialized in agent-negotiation, and this negotiation turns out to be very similar to an open reverse auction, but it can also be turned into an open English auction-like negotiation The interesting fact here is that the auction method incorporates strictly bilateral negotiations between bidders themselves, the lowest bidder (in case of the reverse auction) continues to negotiate with subsequent bidders until a new, lower bidder is found. This mechanism continues until the lifetime of the negotiating agent has expired and a result is announced.

Finally, P8 claims that this is the first all-round multi-purpose auction protocol. In fact, it can support a variety of auction types through slight amendments of the maximum bid calculation (which is not specified explicitly) and the type of announcements that are made by the auctioneer.

3.2 Method and Level of Complexity

The level of complexity in this context is meant to describe the amount of complex mathematics that has gone into the protocol to make it work. This has an impact on the implementation in so far as certain mathematical skills by the implementers are required and maintenance of some implementations can be demanding if the underlying methodologies are complex.

P1 uses simple public key cryptography as well as some computations based on rest classes, which is not exactly rocket science.

P2 uses a fairly complex combination of old and new cryptographic methods, namely public key cryptography, threshold secret sharing schemes and verifiable signature sharing.

In terms of complexity, P3 is probably among the more complex protocols, using secure distributed computations based on operations on polynomials over a finite field.

P4 is medium complex, using secure additions at the core of its algorithm.

By its description, P5 is by far the most complex and difficult to understand. It uses homomorphic encryption with Φ-hiding assumption which is not trivial.

P6 uses public key cryptography for chains of bids with time stamps on the bids that are actually provided by an online notary. In terms of complexity, this protocol falls into the medium-complex bracket since the determination of the chain elements is not that complicated.

P7 is simply working on the basis of public key cryptography and certificates; therefore the underlying technology is well understood. However, since this protocol is based on agent negotiation, the realization of agents may be a non-trivial task.

Also P8 is only based on public-key cryptography primitives. It does not require any technology beyond that, therefore, it can be considered not very complex. In case of the variant that can enforce time-restrictions, the protocol requires a trusted clock (e.g. a hardware addition to the client machine in the form of a trusted co-processor).

3.3 Security

Security describes a fairly wide number of properties in a protocol. It means for instance that tampering or replay of messages by a third party, or corruption/loss of messages should not cost more than some additional computation to either party. It also means that a protocol follows its deterministic path, i.e. that the auction closes at the correct moment and that the correct amount of money is collected from the correct

person. Furthermore it means that participants in the auction must be accountable for their actions. Generally, it means that the auction executes securely.

P1 is only a building block of a larger auction service. It is secure in its core algorithm, i.e. it is not possible for the participants to know about each other's bid amounts. However, the overall security pretty much depends on how the auction based on this core algorithm is designed. Since the participants themselves have to take care of the comparison of their bids, there are certainly possibilities for cheating, e.g. by not committing to ones' bid, i.e. claiming the own bid is lower. From this perspective, the protocol is not very sophisticated and secure.

P2 has been designed with a lot of security features in mind. It ensures a variety of things. The bidding period eventually closes, but only after a correct auction server decides that it should be closed. There is at most one winning bid per auction, dictated by the publicly known rule applied to the well-formed bids received before the end of the bidding period. The auction service collects payment from the winning bidder equal to the amount of the winning bid. Correct losing bidders lose no money, and only the winning bidder can collect the item bid upon. It is probably fair to say that this protocol is one of the better ones in terms of security, since many angles are covered in it.

P3 ensures the accountability of bidders via their signatures on the bids. It also allows for accountability of the auctioneer. Coalitions of auctioneers smaller than a certain size cannot determine any information about the bids; only a majority of correctly operating auctioneers can do that. P3 can be considered very secure.

In the description of P4, no particular mention is made about security as such. Therefore, the behavior of the protocol with regards to message tampering etc. is uncertain.

P5 looks better than P1 from the security perspective, it remains unclear what effect the loss or tampering of messages would have on an auction outcome with regards to the proposed two-server auction implementation.

P6 has a few security mechanisms that are introduced due to the presence of an online notary. First of all, only authorized clients can submit a bid. When committed to bids, and to the bid order, the auctioneer cannot discriminate between bids based on the bidder or bid amount. Only bids made within a reasonably small interval of the present time can be reordered. The auctioneer cannot refuse a committed bid and bidders, and sellers can verify the correct operation of the auction in an acceptable amount of time. It is interesting to note that this protocol is the only one that explicitly distinguishes between the seller and the auctioneer. Additional features to add security that are mentioned in the protocol is the deployment of a concealed routing mechanism and a certified delivery service if this should become necessary.

P7 is mostly geared at ensuring the security of the automatic agents, or, in other words to make misbehavior of servers against agents at least provable. The guarantee for integrity of the protocol is through a list of inter-linked negotiation results that ensure that agents cannot suffer from replay attacks or simply from termination by a malicious server.

P8 is proven to be secure against passive and active attackers (eavesdroppers and agents inserting fake messages into the message flow) using BAN logic. However, it

has to be stated here that the protocol does not safeguard against all misbehavior types of the auctioneer, e.g. it is not ensured that the auctioneer indeed chooses the maximum bid amongst the bids he has received, especially when it comes to the modification of the protocol to a secret-bid auction where the highest bid may need to be hidden to ensure privacy. Nothing is there to stop the auctioneer then to collude with one of the bidders to fix the sale at a lower price with nobody else knowing about it.

3.4 Anonymity

Anonymity means that nobody (ideally including the merchant) should know the true identity of the customer involved.

P1 is not designed with the keeping of anonymity in mind at all. The only way of ensuring anonymity for P1 would be to use pseudo-anonymous mechanisms, e.g. fake identities for the bidders. This is fact is true for most of the protocols.

In P2 the identity of a correct bidder and the amount of its bid are not revealed to any party until after the bidding period is closed. However, the anonymity is then lifted in that very moment, which may not be desirable in some instances. The authors show an extension to the protocol with which anonymity can be achieved.

P3 can guarantee anonymity only if anonymous (or pseudo-anonymous) public keys are used.

P4 does not mention anonymity at all. In P5, the server has to learn about the identity of the customers.

Due to the type of the auction in P6 which is English, by definition the bidders know about all bids in every round of the auction, and anonymity is explicitly mentioned as not being one of the goals of the protocol.

In P7, anonymity is not an issue. Trade agents are the negotiating parties, and these are identifiable through the server they originate from.

P8 is formally proven to maintain anonymity of the bidder (in fact that is a pseudo-anonymity via a acquired pseudo-identity).

3.5 Privacy

Privacy means that no third party should be able to know details about the transaction that has taken place (payment amount, product details etc.).

Again, P1 is only part of a full auction mechanism. But even if a clever mechanism were found that would perform the bid comparisons in an order that is not visible to the bidders, nevertheless would the last loser of the bid comparison know about what his bid was at least (since he was the last survivor together with the winning bidder). However, the key feature of the protocol is that the bidder does not get to know details of the other bidder's amount, in that sense there is privacy.

P2 reveals the bid size at least to the auction servers once the auction is closed to kick off its comparison algorithm. This means to say that at least at this moment could a malicious auction server pass information to outside parties who might be interested in this.

P3 is designed to determine a selling price without revealing any of the bid values other than the second highest of course.

P4 guarantees that no auction bid is going to be revealed except for the winning bid.

One of the major design features of P5 is preserving the secrecy of the bids. Not even the auction server learns about the content of the bids, it can only compute a partial order of the bids.

Due to the nature of the auction type in P6 (English auctions), bids are purposely made public to all participants of the auction.

Since the negotiations are of bilateral nature without the deployment of any millionaires'-type protocol in P7, the bids are of course not kept secret from negotiation partners.

In P8, the bid amounts remain a secret between the bidder and the auctioneer.

3.6 Payments: Atomicity and Non-repudiation

Atomicity means that under all circumstances, the transaction must either go to completion (the right goods are exchanged for the right amount of money) or be aborted.

Non-Repudiation in this context means it is not possible to default on a payment on the grounds of denial of the commitment.

P1 does not take into account atomicity and non-repudiation at all since the protocol does not extend to the actual exchange of money for the goods; it merely is focusing on the actual bid comparison process.

In P2, the bidders pay the money up-front, therefore the servers are already in possession of the electronic coins in a way from the beginning of the bidding period, however in encrypted format. As soon as the electronic coins are decrypted, the servers can deposit them into a bank account. This does not necessarily force the auction servers to deliver the goods, though. However, as soon as the winning bid is determined, the coin can be decrypted. In that sense the protocol preservers non-repudiation.

Settlement is not mentioned as part of the original protocol as described in P3. However, the authors suggest a similar deposit scheme like the one used in P2 to ensure non-repudiation.

P4 ensures non-repudiation, i.e. no winner can repudiate his bid. However, no deposit mechanism is mentioned here.

P5 behaves like P1 with regards to non-repudiation, i.e. no special mechanisms are built into the protocol in order to avoid repudiation. Again, there is no mention about atomicity, since the settlement is not part of the protocol.

P6 ensures non-repudiation on the side of the bidder through the chain-mechanism, and non-repudiation on the side of the auctioneer using the online notary mechanism.

The description of P7 does not cover the payment phase and does not make any comment on non-repudiation or atomicity. With regards to non-repudiation, we can assume that this is enforced simply due to the fact that the agents sign their bids.

In P8, the message exchange is proven to maintain atomicity and non-repudiation; this is ensured since all messages are signed.

3.7 Efficiency

Efficiency means that mathematics within a protocol should not be computationally complex, but should scale ideally logarithmically or less with the increase of auction size (in terms of number of bidders or number of possible bids).

We only repeat here what has been said in the papers describing the protocols, since a separate evaluation of the complexity of these protocols is beyond the scope of this thesis.

One of the criticisms about P1 that have been stated e.g. in [1] is that it operates extremely inefficiently in its comparison algorithm.

The authors of P2 have run a few test auctions with their protocol to determine the efficiency of their protocol. Their results show that the auction speed increases linearly with increasing numbers of bids.

P3 is described by the authors as non-trivial in terms of complexity; they mention that it is certainly only applicable for big auctions running on big servers.

The complexity of P4 is highly dependent on a variable k that determines the structure of the bidding interval. No remark is made about the actual performance of the protocol, though.

The core design goal of P5 besides privacy is the efficiency, since in fact it recreates P1 in a more efficient manner. Nothing is mentioned, though, about the actual speed of the protocol.

There is no analysis or discussion about the efficiency of P6; it depends partly on the network performance, the actual public/private key methodology and other factors, so we cannot give a statement about the efficiency here. The same argument holds for P7, no statement can be made about efficiency.

Since P8 uses only very simple cryptographic primitives, an implementation would be quite efficient, since it would not be computationally demanding.

3.8 Incorporation of Timing Issues

Timing issues mean that the protocol should ensure that certain time constraints that have been imposed on the auction process should be adhered to. One such example would be the requirement that the auctioneer (or the seller) must deliver the product (provided it's electronic) within a maximum amount of time to the highest bidder. The only protocol that allows for such a possibility is P8 with an extension. The method is a slight amendment to P8 and the installation of a trusted clock at the client's site (the issue is whether the auctioneer trusts this clock as well, which has not been discussed in [10]). No other protocol provides this possibility.

3.9 Ease of Implementation

Ease of implementation refers to the feature of a protocol that makes it easy to take it as is (i.e. as it is described in the literature) and install it as an online auction system. One of the factors that determine this feature is the complexity of the mathematics underlying the protocol; another one is how complete the protocol describes an auction mechanism. A third factor could be e.g. the requirement for additional services for the protocol to function, e.g. trusted third parties or online notaries.

From this perspective, P1 is medium-difficult to implement since, despite it being based on a rather simple algorithm, it does not feature the mechanics of full auction service, e.g. settlement function and goods delivery steps are missing.

P2 has more complex mathematics and requires a distributed auction mechanism that requires communication protocols between the participating servers. From that perspective, it is more complex to implement.

Similar to P2, P3 is a rather complex protocol that requires multiple servers and therefore is expected to be complex to implement.

P4 in the other hand appears rather straightforward to implement since it runs a rather medium-complex algorithm on a single server only.

In terms of ease of implementation, P5 would be a bit more difficult to implement than P1 since its underlying algorithm is much more complex than that of P1. Furthermore, it requires two servers to act in tandem.

Since a few other things are required besides the protocol implementation as such (e.g. online notary, certified delivery services etc.), the implementation of P6 is not necessarily very easy and straightforward.

P7 is probably not that straightforward to implement either as it is based on the assumption that trading agents are conducting the auction steps. The underlying cryptographic methods are quite easy, however, the trade agent context may make the implementation non-trivial.

The implementation of P8 should be rather straightforward since the protocol describes the entire auction from the advertising phase for the good to the settlement, so no additional mechanisms have to be designed.

4 Conclusion

We have seen the various protocols and each shows certain ideas and approaches to particular problems within online auctions in varying complexity and detail. Some are approaching the subject by implementing merely clever comparison protocols. Others focus their attention on the secure deployment of agents for the purpose of finding the cheapest product. Still others look into how to make sure that multiple auctioneers (or auctioneer servers) do not misbehave (alone or collusive), and if they do, how the effect can be kept to a minimum. Furthermore, not all protocol descriptions as they appeared in the literature provide the same level of detail with regards to actual implementation possibilities, efficiency in various environments etc.

Therefore, if we assume that the problem at hand would be to implement an online auction house to support a variety of auction types, we should take care of possible time-constraints, be as secure, anonymous, etc. as possible. We should allow the bidder to remain anonymous if she wishes to. All is not an easy task simply looking at the protocols that have been developed so far. In fact, conclusive answer can be given to which protocol is the ideal one for such an implementation.

The answer to this problem would probably be a combination of the various technologies and methods that have been deployed in the protocols. Of course, such integration of technologies and methods is not a trivial task. There is still a lot of scope for future development in this area of fusion of cryptographic and other technologies, for example, a distributed system synchronization, that enables online auction software among possibly other things to be secure, flexible, technically sound, yet efficient and easy to implement.

Undoubtedly, in order to create online auction servers with good security, anonymity, and following the user demands, more work has to be done to fuse some of the ideas and technologies that have been published and create a new type of online auction platform that can survive in the world of B2B electronic auctions. The trend is certainly visible, and the introduction of automatic negotiation agents that perform negotiations between companies on these companies' behalves certainly requires a more stringent security model for electronic negotiation platforms in general and electronic auction platforms in particular.

References

1. Christian Cachin. **Efficient Private Bidding with an Oblivious** Third Party. Proceedings of *the 6th ACM Conference on Computer and Communications Security* (CCS 99). pp.120-126. Singapore, ACM. 1999
2. Matthew K. Franklin and Michael K. Reiter. **The Design and Implementation of a Secure Auction Service**. Proceedings of *the IEEE Symposium on Security and Privacy*. 1995
3. Michael Harkavy, J.D. Tygar and Hiroaki Kikuchi. **Electronic Auctions with Private Bids**. Proceedings of *the 3rd USENIX Workshop on Electronic Commerce*. Boston, Massachusetts. September 1998
4. Manoj Kumar and Stuart I. Feldman. **Internet Auctions**. Proceedings of *the 3rd USENIX Workshop on Electronic Commerce*. Boston, Massachusetts. September 1998
5. David Lucking-Reiley. **Auctions on the Internet. What's Being Auctioned, and How?** Department of Economics. Vanderbilt University. 14th August 1999
6. Philip MacKenzie and Jeffrey Sorensen. Anonymous Investing. **Hiding the Identities of Stockholders**. Proceedings of *3rd Financial Cryptology Conference*. Lecture Notes in Computer Science #1648. pp. 212-229. Springer Verlag. 1999
7. Rajiv Mehrotra. **Online Auctions: Just Another Fad?**
8. H. Nurmi. **Cryptographic Protocols for Auctions and Bargaining**. In: Results and Trends in Theoretical Computer Sciences. Lecture Notes in Computer Science #812. pp. 317-324 Springer. 1994
9. Srividhya Subramanian. **Design and Verification of Secure E-Commerce Protocols**. Dissertation at the Ohio State University. 1999

10. Srividhya Subramanian and Mukesh Singhal. **Real-Time Aware Protocols for General E-Commerce and Electronic Auctions Transactions**. Proceedings of *ICDCS Workshop*, June 1999
11. Stuart G. Stubblebine and Paul F. Syverson. **Fair On-Line Auctions without Special Trusted Parties.** Proceedings of *3rd Financial Cryptology Conference*. Lecture Notes in Computer Science #1648. pp. 230-240. Springer Verlag. 1999
12. J.D. Tygar. **Atomicity in Electronic Commerce**. Computer Science Department. Carnegie Mellon University. Pittsburgh, PA.
13. Dietmar Waetjen. **Kryptologie**. *Lecture Notes*. Institut fuer Theoretische Informatik. Technische Universitaet Braunschweig. Oktober 1999
14. Xiao Feng Wang, Kwok-Yan Lam, Xun Yi, C. Q. Zhang, and Eiji Okamoto. **Secure Agent-Mediated Auction-like Negotiation Protocol for Internet Retail Commerce**. Lecture Notes in Computer Science #1652. pp. 291-302. Springer. 1999
15. A.C. Yao. **Protocols for Secure Computation**. *23rd Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press
16. Xun Yi, Xiao Feng Wang, Eiji Okamoto and D. Frank Hsu. **A Secure Auction-like negotiation Protocol for Agent-based Internet Trading**. pp. 197-203.
17. Makoto Yokoo, Yuko Sakurai and Shigeo Matsubara. **The Effect of False-name Declarations in Mechanism Design: Towards Collective Decision Making on the Internet.** pp. 146-153
18. Reyne Haines, Arden Snyder, and Connie Swain. **The Auction Book – 1999 Guide to Online Auctions**. GWB publishing, Cincinnati, Ohio, 1998

Author Index